Between Non-ventricular Rhythms, Ventricular Tachycardia and Ventricular Fibrillation in the Electrocardiogram

Alwan, Yaqub

Awarding institution:
King's College London

Methods for Automatic Differentiation Between Non-ventricular Rhythms, Ventricular Tachycardia and Ventricular Fibrillation in the Electrocardiogram

Yaqub Alwan

A thesis submitted to King's College London for the degree of

Doctor of Philosophy

Department of Informatics

King's College London

10th June, 2016

Towards the advancement of humanity, science, and critical thinking.

Acknowledgements

First and foremost, I must express my deep gratitude to Professor Zoran Cvetković for having accepted me to the PhD programme, and for his continuous support and guidance throughout. I am also grateful to Dr. Michael Curtis, for having suggested the topic of research which allowed me to realise my dream of making a contribution to the field of medicine. I am extremely grateful for the support and input of both my supervisors, without which this journey would not have been possible.

Words cannot express enough the importance of the roles of my wife, Sana, her mother, Saleema, and my own mother, Zara, for their unwavering support during what can only be described as the most challenging period of my life. Thank you, Sana, for keeping my head above the surface when I felt I could not.

Truly missed, will be the company and time spent with my CTR colleagues. Special thanks go to Pauline (Paul), Nooblies (Adnan), Alexandre, the Goats (Christoforos and Giorgos), Omar, and Enzo for all our entertaining, insightful and off-topic moments. My four years at CTR have been a treasure because of you.

Finally, I would like to acknowledge the funding support of the School of Natural and Mathematical Sciences, King's College London.

Abstract

This thesis is concerned with the analysis and development of methods for simultaneously distinguishing between non-ventricular rhythms, ventricular tachycardia and ventricular fibrillation in the electrocardiogram. A realistic experimental framework for assessing methods was developed that does not over-estimate accuracy of investigated methods, and descriptive statistics were used for reporting results of experimental simulations. The methods developed were tested against recent studies in the literature. The developed methods introduced high dimensional feature spaces for reducing the amount of information discarded, and the best method achieved 30% reduction in median error rates by combining multiple feature spaces, directly and in a hierarchical fashion, and through incorporation of rhythm context from past observations, as opposed to conducting analysis on the currently observed segment alone. The research conducted has not solved the problem of differentiating between non-ventricular rhythms, ventricular tachycardia and ventricular fibrillation entirely, and remains an open problem for research. Through the development of methods in this thesis and observations made, many more avenues are proposed for improving automated rhythm diagnosis in the electrocardiogram.

Contents

List of Figures

List of Tables

Acronyms

AED automated external defibrillator. 20, 43, 135, 140

AF atrial fibrillation. 27, 52

AHADB American Heart Association Database. 51–53

AICD automatic implantable cardioverter defibrillator. 19, 25, 33, 43, 135, 140

AUC area under the curve. 48, 63

CRF conditional random field. 118, 119, 122–124, 127–129, 133, 134

CUDB Creighton University Ventricular Tachyarrhythmia Database. 51, 52, 137

ECG electrocardiogram. 15, 17, 19–31, 51, 52, 55–57, 59, 60, 63–65, 71–73, 83, 89–92, 96, 99, 115, 117, 120, 128–133, 136, 139

EDB European ST-T Database. 51, 52

EMD empirical mode decomposition. 28, 33

HMM hidden Markov model. 118–123, 128, 133

IMF intrinsic mode function. 28

LCE local context ensemble. 91, 92, 99–101, 104, 108–111, 117, 132

LDA linear discriminant analysis. 30, 35, 37, 40, 58, 59, 73, 75, 81, 96, 98, 134

MEMM maximum entropy Markov model. 121, 122, 128, 133

MITDB MIT-BIH Arrhythmia Database. 51, 52

NVR non-ventricular rhythms. 17, 20, 22–26, 28–30, 32, 43, 51, 53, 54, 56, 58–60, 63, 66, 67, 75, 80–85, 92, 100, 102, 104, 106, 108–111, 116, 120, 124, 125, 127, 130–133, 135–137, 139, 140

PC principal component. 65, 66, 73, 75

PCA principal component analysis. 49, 57, 65, 66, 75, 82, 131, 132

QDA quadratic discriminant analysis. 36, 37, 40, 59, 73, 75, 81, 82

RBF radial basis function. 39, 69, 70, 72, 73, 75, 79, 81, 82, 97, 99, 110, 124, 132

ROC receiver operating characteristic. 48

SR sinus rhythm. 17, 24, 27, 28, 130

SVM support vector machine. 33, 38–45, 49, 50, 57, 59, 67–69, 72, 73, 75, 79, 81, 82, 85, 87–97, 99, 108–110, 115, 116, 124, 127, 131–133

VF ventricular fibrillation. 17, 19, 20, 22–32, 43, 51–55, 57–60, 63, 64, 66, 67, 71, 75, 80–85, 92, 100, 102, 104, 106, 109–112, 114–116, 120, 124, 125, 127, 130–133, 135–137, 139

VFDB MIT-BIH Malignant Ventricular Arrhythmia Database. 51, 52

VT ventricular tachycardia. 17, 19, 20, 22–32, 43, 51–56, 58–60, 64, 66–68, 75, 80–85, 92, 100, 102, 104, 106, 108–112, 114–116, 120, 124, 125, 127, 130–133, 135–137, 139

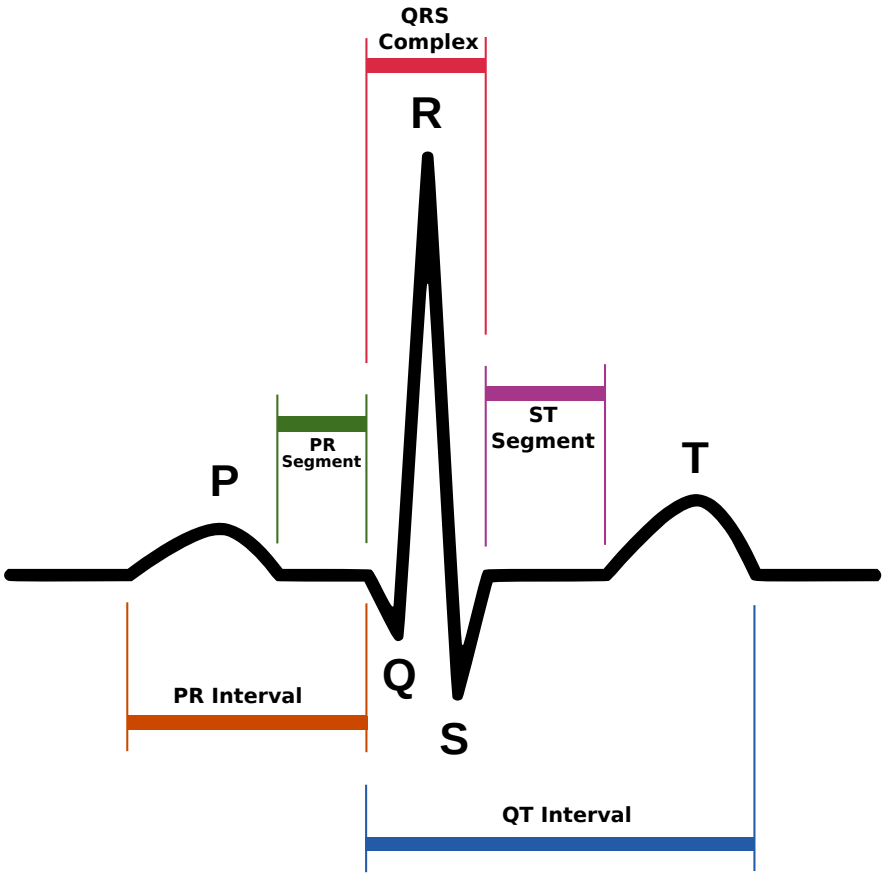
Chapter 1

Introduction

Cardiovascular disease is the leading cause of death in middle and high income countries, and among the top ten causes of death in low income countries according to the World Health Organisation [1]. Development of effective drug treatments that prevent cardiac arrhythmias is therefore a high-priority challenge for modern pharmacology. For the development of such treatments it is crucial to have a clear understanding of what distinguishes different forms of arrhythmia, and based on that, establish their precise definitions. Working towards better automated arrhythmia detectors is a particularly important task.

1.1 Electrical conduction system of the heart

In a healthy heart, an electrical impulse will originate at the sinoatrial node and stimulate the atria, causing their contraction and forcing blood from the atria to the ventricles. This impulse is seen as the *P wave* on the surface electrocardiogram (ECG). The impulse travels through some electrical pathways known as internodal tracts to the atrioventricular node, which is positioned in the walls between the atria and ventricles, and is responsible for delaying the impulse



Credit: Public domain, available at

`http://en.wikipedia.org/wiki/QRS_complex\#mediaviewer/File:SinusRhythmLabels.svg`

Figure 1.1: Illustration of a single normal heart beat with annotations of all the morphological features typically expected in the ECG. The diagram is idealised, and in practice observed ECG beats have a high variability due to many factors such as interference from noise sources, e.g. muscular activity or patient breathing, and also due to disease or underlying conditions.

for a short period of time to ensure the ventricles contract after the atria. This delay is seen as the *PR segment* on the ECG. The impulse then travels through the bundle of His to the Purkinje fibres and left and right bundle branches along the surface of the ventricles, causing them to contract and forcing blood out through the aorta and pulmonary artery. The *QRS complex* on the ECG corresponds to this electrical activity causing ventricular contractions. Finally, another, usually small peak, known as the *T wave*, corresponds to re-polarisation of the ventricles, the period during which there is no surface electrical activity. Figure 1.1 shows an annotated ECG example corresponding to a single cycle of a heart beat.

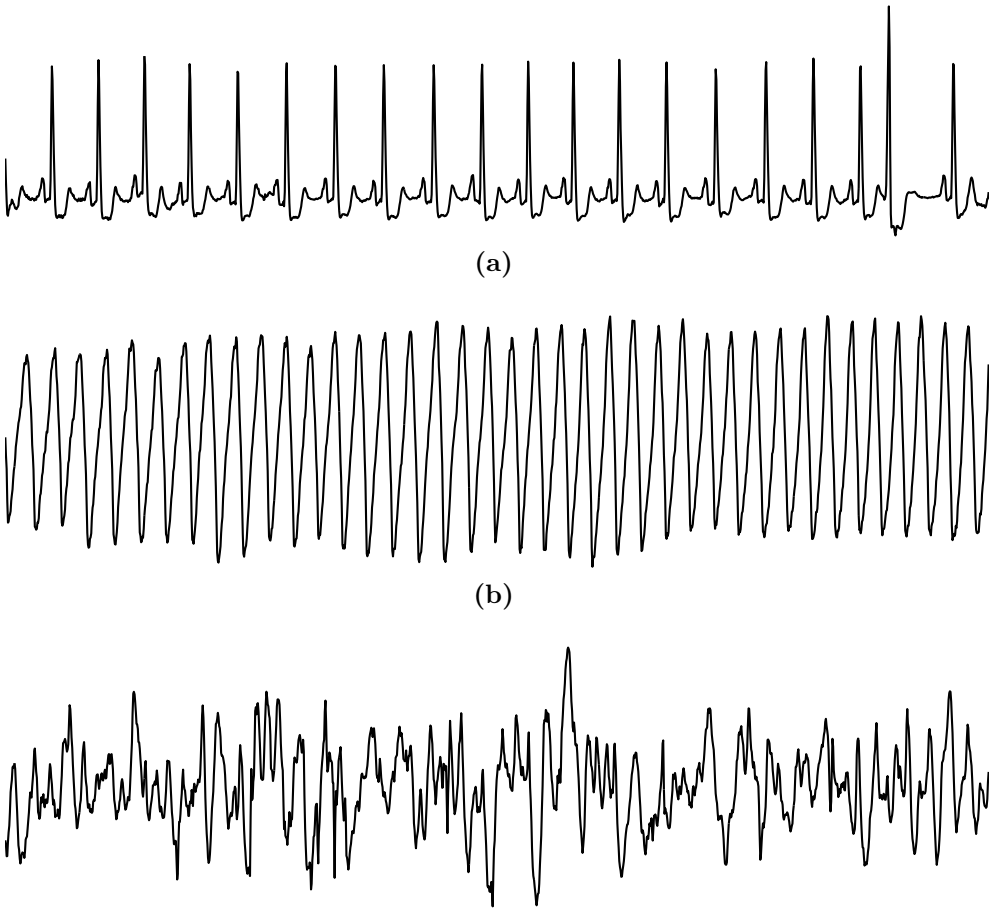
There is a large amount of variability in the morphology of the electrical conduction when observed via the surface ECG, from patient to patient. Additionally, sources of noise such as patient breathing and muscular activations are present. Disease or underlying conditions also alter the morphology of the surface ECG. There are three main categories of rhythm which are of interest, these are:

1. Non-ventricular rhythms (NVR), composed mostly of sinus rhythm (SR) and all other rhythms which are not ventricular arrhythmias. These rhythms almost always manifest on the surface ECG with the morphological features in Figure 1.1.

2. Ventricular tachycardia (VT), which is one form of ventricular arrhythmia, and may be life threatening. These rhythms usually manifest on the surface ECG with wide QRS complexes, no pauses, and no P waves, although T waves can occur.

3. Ventricular fibrillation (VF), which is the other form of ventricular arrhythmia, and is almost certainly lethal. These rhythms manifest on the surface ECG with no discernible QRS complexes, or other morphological features.

A 15 second long example of each of these rhythms is shown in Figure 1.2.



(a)

(b)

(c)

Figure 1.2: 15 second long examples of each of (a) SR, (b) VT, and (c) VF

1.2 Diagnostic difficulties in the ECG

Although unequivocal VF, sustained and lethal, is incontestable in surface ECG recordings, clinicians differ about the diagnosis of transient ventricular tachyarrhythmias, with experts in a landmark report unable to agree on whether VF, polymorphic VT or torsades des pointes best described a range of human tachyarrhythmias in a blinded test of ECG records [2]. Another study noted the difficulty in assigning appropriate rhythm categories when relabelling existing databases for evaluation of an automated classifier [3]. In fact, a couple of studies have used the notion of a VT-VF [4,5] category which is not a well accepted or defined notion, as a tacit admission of the difficulty discriminating between VT and VF. Given that mechanisms of these tachyarrhythmias may differ [6], and responses to drugs may vary from benefit to proarrhythmia depending on the type [7], errors in diagnosis are potentially hazardous. From a therapeutic point of view, being able to properly differentiate between VT and VF is very important since they respond to interventions differently, and VF is usually lethal, while VT is often not. In particular, per patient programmable automatic implantable cardiovertor defibrillators (AICDs) attempt to differentiate between VT and VF, but still deliver unacceptably high rates of inappropriate shock treatments [8].

Although VF is usually self sustaining in humans, but not always [2], it is commonly transient in animal models, especially in mouse, the favoured species for gene modification research [9]. To allow preclinical research to be translatable, guidance was proposed [10], and recently updated [11], for discrimination between VF, including brief and transient VF, and other polymorphic ventricular tachyarrhythmias. The definition however, is not readily transformed into an algorithm for automatic rhythm classification. Therefore in this thesis, automatic methods to discriminate between VT and VF are explored, given only

some surface ECG recordings and rhythm annotations. This is explored assuming the context of automated external defibrillators (AEDs), where usually only a single surface ECG lead is available, per patient customisation is not possible, and identifying NVR properly is essential to avoid delivering unnecessary treatment. Most of the previous approaches only consider NVR vs VT/VF, which in real life application to an AED would result in defibrillation shock treatment for VT, despite the preferable treatment being electrical cardioversion which delivers less energy. On the other hand, drug treatments may be available to individuals considered to be at risk, however, the drug discovery process is hampered by the inability to diagnose rhythms correctly with automated processes.

1.3 Thesis structure

The thesis contains six chapters. Chapter 2 provides some terminology unification, and surveys existing methods for ventricular arrhythmia diagnostics which fall into a variety of categories. The limitations of these methods are discussed and form the basis of various arguments and design choices throughout the thesis. Machine learning concepts are introduced, with particular focus on methods which are commonly used in the thesis. Particular attention is paid to classification methods, experimental procedures and assessment metrics. Finally the databases used in the thesis are with statistics provided on rhythm prevalence and details on database characteristics.

Then, preliminary experiments are explored and conducted in Chapter 3. Details are provided on the developed experimental framework, with disclosure of as many details as possible to allow for reproduction of simulation experiment findings. A large variety of representation spaces from the literature, and and new representation spaces, are considered, as well as a variety of classification algorithms and parameters, with the goal of reducing the amount of methods

requiring testing by keeping only those showing the highest promise.

This forms the basis for the next development in the thesis, classification using ensemble methods, which is presented in Chapter 4. The ensembles are constructed in a temporal fashion, in an attempt to exploit correlations among sequences of observations, which is an approach not seen previously in ECG diagnostic methods. Additionally, hierarchical type ensembles are introduced and constructed in order to build upon the strengths of different feature spaces. This approach has also not seen use previously in ECG diagnostics.

Building upon the observation in Chapter 4 that ensembles formed temporally provide a useful and non-trivial improvement, an attempt is made to utilise existing methods designed precisely for exploiting observation sequence interactions, using the conditional random fields technique. This is performed under time constraints, so only a preliminary study is performed, due to particularly long training times for this class of methods.

Finally, the thesis is summarised in Chapter 6. The contributions and achievements are discussed, as well as the limitations of the conducted studies. Attention is paid to mislabelling in the ECG databases, with recommendations on how to improve the situation, and sequential learning methods are discussed briefly, with recommendations on methods to be considered for future research and some insight into why they may perform well.

1.3.1 Contributions made in this work

The overall contributions of the work presented in this thesis are listed. These are

1. Development of a realistic experimental assessment framework that is not optimistic in accuracy estimates

2. Use of descriptive statistics when presenting experimental results, rather

than potentially misleading significance tests on single aggregate values. In particular, probability density estimates of accuracy metrics were built

3. Consideration of ECG diagnostics in true multi-class settings, i.e. no proxy rhythms are used, nor are rhythms that are difficult to tell apart merged into single categories. The scenario considered is NVR vs VT vs VF, consistently throughout the thesis

4. Consideration of high dimensional feature spaces, and rather than following the trend towards feature reduction, consideration of feature expansions via temporal ensembles

5. A true assessment on how the methods developed generalise to new patients, by not exposing test patients to any part of training or parameter tuning

6. Consideration of systematic feature reduction via principal component analysis

7. Combining of feature spaces according to their apparent strengths in order to develop an overall stronger classifier

8. Development of methods to exploit temporal interactions for classification in the ECG

9. Brief consideration of principled methods for exploitation of temporal interactions

10. Identification of database mislabelling, which may be detrimental to development of further improvements to classification algorithms. Recommendations are provided on how this situation can be improved

Chapter 2

Background and methods

Most prior studies that perform rhythm classification in the ECG evaluate binary classification tasks. Some of these are reviewed, along with methods that perform three-way classification between NVR, VT and VF. One of the main problems with the majority of methods proposed in the literature is that the claims made by original studies are usually unsupported by further studies.

This chapter will consist of three parts. First, methodologies presented in previous studies will be discussed, and a summary of shortcomings is produced. Then, a selection of techniques from machine learning literature are reviewed. This includes classification learners, metrics for classification assessment, and methods for conducting studies in a proper fashion without introducing bias. Finally the data used for the work in this thesis are described and examined briefly.

2.1 Terminology

In this thesis, a variety of terms are used, sometimes interchangeably. They are covered here in one place for easy reference.

- The terms NVR and SR are used interchangeably unless specified, despite not being interchangeable in the general case. This is mainly due to the vast majority of NVR actually being SR, both in practice, and in prevalence in the databases used.

- Feature space, representation space, feature vector, transformation are all terms used to describe some kind of transformation, or map, that has been applied to raw data to produce a fixed sized vector.

- Training and learning are used interchangeably to mean processes that estimate functional relations between data.

- Testing and inference are used to mean processes that evaluate data points using an estimated function, often by producing a probability estimate of a given outcome.

- Diagnosis, differentiation, and classification are all ways to describe tasks where the goal is to categorise observations.

- Window, segment, and observation are used to mean a portion of the ECG being considered for analysis.

- Category, class, group are all used interchangeably to refer to a distinct, defined group within data.

2.2 Literature survey and prior art

Many studies are presented in the literature, for differentiating between NVR and VF (sometimes combined with VT), while others attempt to perform classification between all three categories. The features computed by these studies for these purposes are broadly categorised into three main groups; (i) morphological,

(ii) spectral and, (iii) dynamical/complexity features. Some of these approaches are discussed briefly, and their limitations are elaborated. The transformations or computed featured are focused on, since it will be shown that a majority of approaches utilise ad-hoc decision making processes which are not grounded in statistical learning theory.

2.2.1 Methods based on morphological feature extraction

First, some methods performing rhythm detection using representation spaces computed from ECG morphology are described.

For classification in AICD devices, a rule-based algorithm was developed which attempts to differentiate between NVR, slow VT, fast VT and VF [4]. Cardiac deflection detection is performed by measuring slope and performing thresholding. After the detection of a deflection, a blanking period is applied where new deflections are not detected. When the next deflection not in this blanking period is detected, it is preliminarily categorised based on the time since the previous deflection. Further categorisation is then performed by analysing the previous 24 deflections according to a set of predefined rules. This algorithm benefits from the ability to make per-patient customisations in order to tune the decision rules for the arrhythmia categories, which is important in the AICD setting.

Cross correlation peak amplitude and peak widths [12] are computed between a prior window of ECG and the current window. Then, the standard deviations of the peak widths, and peak amplitudes respectively, are computed and used for classification between NVR, VT and VF using an ad-hoc decision system derived from the distributions of both parameters, which are obtained using the entire data set.

Threshold crossing interval [13, 14] is computed from a window of ECG and

used for classification. The current window of ECG is used to compute a threshold, and the sequence is converted to a binary sequenced based on whether the ECG is above or below this threshold. The number of times the threshold is crossed is counted, and boundary effects are corrected for using the previous and next window of ECG [13], or by using a longer window [14]. Then, for classification between NVR, VT and VF, a sequential hypothesis test is used [13], and for classification between no VF and VF, support vector machines are used [14].

Threshold crossing sample count [15] aims to improve upon the threshold crossing interval algorithm, by noting a weakness in that the number of crossings are computed using the value of the ECG sequence, rather than the absolute value. To deal with boundary effects, a tapered cosine window is applied to the ECG segment. The amount of the ECG sequence above the absolute value of the threshold is the counted, instead of the number of crossings. The sequential hypothesis test is replaced by a simple ad-hoc detection scheme formed by inspecting probability distributions of the computed parameter.

A signal comparison algorithm [16] is developed alongside a comprehensive review of a variety of other techniques for arrhythmia detection. A number of templates, segments of each NVR, VT and VF are selected. An incoming window of ECG is analysed to find important local maxima. These are used to generate reference sequences based on the templates, of the same length as the episode to be analysed, and the ℓ^1-norm between these references and the input ECG are computed. Again, ad-hoc decision rules are applied to the computed values in order to decide whether VF is present or not. An in depth comparison with a variety of other algorithms is also presented, including some discussed here, as well as other key algorithms such as the standard exponential algorithm and modified exponential algorithm.

2.2.2 Methods based on spectral feature extraction

Methods for extracting features based on morphology have some limitations. They often require complex logic in order to deal with expected variations in the ECG morphology. On the other hand, simpler techniques may be used, but are likely to be overly simplistic, e.g. threshold crossing interval. Feature extraction using spectra, or operations that manipulate the spectra, e.g. linear filtering, can potentially operate better, due to invariance to time shifts. These operations can be performed without the need for complex algorithms to align observation windows or search for landmark points in the ECG.

The band-pass filtering approach computes statistics on the output of ECG segments filtered using an integer band-pass filter [3]. The absolute value of the band-pass filtered ECG, FO, is used to compute three different metrics;

(a) number of output samples in the range $0.5\max(FO)$ to $\max(FO)$,

(b) number of samples in the range $\text{mean}(FO)$ to $\max(FO)$, and

(c) number of sample in the range $\text{mean}(FO)$ to $\text{mean}(FO) + \text{std}(FO)$.

Histograms of these parameters are computed across the entire dataset and are used to develop an ad-hoc decision scheme which makes a shockable decision or non-shockable decision. The data, although publicly available, was re-annotated for this study. It was noted that there are several rhythms present in the databases which are hard to categorise even by experts into shockable or non-shockable categories. Also, as part of this re-annotation, all VF were treated as shockable, but not all VT were treated as shockable. Therefore, VT was being spread across categories.

Higher order spectral techniques [17] computes the bispectrum of an entire ECG record in order to differentiate between records containing only SR, atrial fibrillation (AF), VT or VF. The algorithm is applied per episode, or record,

which although unspecified appeared to be of long duration. The bispectrum is reduced to a single parameter by finding the region containing significant energy, and an ad-hoc classification scheme is used to make a diagnosis.

A feature representation extracted using so-called semantic mining [18] was developed to differentiate between SR, VT and VF. The ECG is treated as a dynamical system with three parameters, the natural frequency ω, the damping coefficient ζ, and the input signal μ. These parameters are all estimated using the spectra of an ECG window and the first to fourth-order difference equations of the ECG. Then, analysis of variance and T-test techniques are used to develop an ad-hoc decision scheme.

Spectral features [19] are computed directly from the Fourier transform of a 5 s window of ECG. The frequency F, corresponding to the peak amplitude in the region 0.5Hz - 9Hz is used to compute first spectral moment, and ratios of areas defined in relation to the reference frequency F. A total of four parameters are computed, and scatterplots of these parameters are used to derive an ad-hoc rule based scheme for discriminating between rhythms with complexes of aberrant morphology (assumed to be VT), VF and imitative artefacts.

The empirical mode decomposition (EMD) is an analysis tool that decomposes a time series into a set of additive components known as intrinsic mode functions (IMFs) and a residual component [20]. The general form is given as;

$$x[n] = \sum_{i=1}^{I} \mathrm{IMF}_i[n] + r[n] \ . \tag{2.1}$$

The spectral entropy of the first IMF is computed, as well as the energy ratio of the first IMF to the remaining signal energy and these computed values are used as features for a naive Bayes classifier to differentiate between NVR, VT and VF [21].

Fourier neural network [22] performs classification between NVR, VT and

VF. However, isolated premature ventricular contractions are placed in the same category as VT, despite actually belonging to the NVR category. The input features to the neural network are computed by finding a QRS complex, and extracting a window of the ECG centered at the QRS complex. Then, the Fourier transform is computed and the power spectrum is found. Finally, the first 5 components after the 0Hz component are used as inputs to a neural network classifier.

2.2.3 Methods based on extracting dynamical, complexity and other features

This section describes a small selection of techniques attempting to differentiate between ECG rhythms by using features that are computed by characterising dynamical or complexity properties of the ECG, and other techniques that do not fall into either the spectral or morphological categories.

Lempel-Ziv complexity measure [23] is used to try to quantify the complexity of an ECG sequence [24]. The Lempel-Ziv complexity measure is defined for finite sequences of symbols, so ECG windows of varying lengths are first preprocessed to transform it to a binary symbol sequence, by comparing it to a threshold learned from the given window. Then complexity measure is computed from the symbol sequence, and distributions for each category of NVR, VT and VF using different window lengths are used to determine a threshold for an ad-hoc decision scheme. It is concluded that window lengths of 7 s or longer are appropriate.

For time series data, a measure of regularity can be computed directly, known as approximate entropy. Sample entropy is a modification of this procedure intended to be more robust to noise contamination of the time series, and less sensitive to the chosen observation length. Sample entropy is computed directly [25] and distributions of this value across a dataset are used to derive some ad-hoc

thresholds for classification between NVR, and combined VT and VF.

For detection between NVR and combined VT and VF, phase spaces are constructed from an 8 s window of ECG. These are constructed in two different variants, phase space shifted [26], constructed using a shifted version of the ECG, and phase space Hilbert [27], constructed using the Hilbert transform of the ECG segment. Then, the range spanned by the phase space is quantised into a 40×40 grid, and the the number of uniquely visited positions on the grid is counted. An ad-hoc threshold is used for classification.

Multifractal analysis is used to differentiate between NVR, VT and VF [28,29]. The basic idea is to estimate the singularity spectrum at various scales and compute statistics or features from these spectra. In one case, the singularity spectra across a range of exponents are integrated, to give a single parameter for classification through an ad-hoc scheme [29]. In the other case, the singularity spectrum is computed at two different scales, and the minimum, mean, and maximum of these spectra are computed and used as the inputs to a neural network [28] classifier.

The wavelet transform is used to derive a feature space for categorisation between VT, a VT-VF category, and VF [5]. Publicly available data are used, but since the data annotations do not support the notion of a VT-VF category, the data are privately annotated. A 2-dimensional representation of the ECG is formed by thresholding wavelet decomposition to find dense regions of energy in the time-scale domain. The representation is formed by the number of these areas, and the distance in time between these areas. This is then classified using a linear discriminant analysis classifier.

A QRS reconstruction [30] technique is used to differentiate between VF and NVR in the presence of chest compression artefacts. The wavelet transform is used to isolate local maxima, which is then used to build a set of dynamic templates. The input is then compared with the generated templates using the

ℓ^1 residue between each of the template's autocorrelation and the template-input cross-correlations. A threshold and some simple rules are used to determine if a QRS complex is present, and if not, VF is detected. The study notes that only rhythms where experts could make a consensus on the rhythm type are included for evaluation.

2.2.4 Limitations of prior work

The studies discussed were a representative example of the variety of methods developed. From examining the details of these studies, a couple of trends are apparent, which are summarised in Table 2.1; a large number of the studies in the literature are using an ad-hoc decision scheme which is not grounded in any statistical decision theory. Moreover, every study listed (and many others not listed) reduced the ECG waveform to a very low dimensional space. In the case of a 250 Hz sampled signal (a popular choice in the literature), even in the best case, 14 dimensions, this corresponds with taking just 5.6% of the data, assuming a 1 s observation interval. Of course, information reduction cannot be so trivially described, however, it is clear that such extreme amount of data reduction must also be discarding relevant information that can potentially be used to improve accuracy in the problem of discriminating between VT and VF. Even for studies which are not trying to differentiate between VT and VF, it would seem that the amount of information discarded is extreme.

Another issue that is apparent in many of these studies are the use of non-public data. Many studies record their own data, or use data from the public domain which is then re-annotated. For a few studies this is necessary and unavoidable, e.g. QRS reconstruction [30]. However, in general, this is problematic because it does not allow subsequent studies to validate the claimed results. Studies also often preselect data which fits the categories to be studied for training

and testing, which can lead to a false perception on the generalisation ability of the method. This is similarly a problem for studies which use the data set the algorithm was developed with to assess the accuracy. Finally, some studies are conducted using a very limited set of data for both development and testing, and again this does not allow for accurate assessment of generalisation.

More recently, some studies were conducted into performing feature ranking and selection from among a set of the most studied features proposed for evaluating a) shockable vs non-shockable diagnostics [31, 32], b) non-VF vs VF diagnostics [31]. Features considered include some of those discussed in 2.2.1, 2.2.2, and 2.2.3, as well as others. One approach performs feature selection by using a genetic algorithm to generate subsets of features based on fitness (classification accuracy) [31]. The other approach combines the outputs of several feature ranking algorithms in order to generate an overall feature ranking [32]. It is pointed out by both these studies that previous studies do not evaluate rhythm detection algorithms using data drawn from previously unobserved patients. Despite episode randomisation being used by some studies, it is not sufficient for evaluating the generalisation ability of detection procedures applied to completely new patients.

Table 2.1 shows for each of the studies discussed which experimental good practices are followed, indicated in each column by a checkmark, and the maximum feature dimension investigated. Almost every study has at least one experimental flaw, and sometimes even multiple. Of the studies that do not have any of the experimental flaws listed, none are studying VT vs VF directly. No study considers dimensions higher than 15, and conclusions of [31,32] recommend using reduced spaces (9, 2, respectively) for the tasks of non-VF vs VF and NVR vs arrhythmias. Approximately half of the studies use only a single feature for discrimination.

Table 2.1: For some prominent ECG diagnostic techniques, a set of experimental best practices are enumerated, and which of these best practices were implemented by each study. Additionally, the dimension of considered feature spaces for each study is reported

Method	Public data	Large sample	Independent test data	No preselection	Decision theory	Feature dimension
Medtronic AICD rules [4]		✓	✓			1
Crosscorrelation peaks and widths [12]				✓		2
Threshold crossing interval hypothesis test [13]						1
Threshold crossing interval support vector machine [14]	✓	✓	✓	✓	✓	1
Threshold crossing sample count [15]	✓	✓	✓	✓		1
Signal comparison algorithm [16]	✓	✓		✓		4
Band-pass filter counts [3]		✓				4
Bispectral analysis [17]	✓					1
Semantic mining [18]	✓					3
Spectral features [19]			✓			4
EMD spectral entropy and energy [21]	✓		✓		✓	2
Fourier neural network [22]			✓		✓	5
Lempel-Ziv complexity measure [24]						1
Sample entropy [25]	✓	✓				1
Phase space shifted [26]	✓	✓		✓		1
Phase space Hilbert [27]	✓	✓		✓		1
Short-time generalised fractal dimensions [28]			✓		✓	6
Singularity spectrum area [29]	✓					1
Wavelet features [5]			✓		✓	2
QRS reconstruction [30]		✓	✓			4
Genetic algorithm feature selection [31]		✓	✓	✓	✓	14
Weighted combined feature selection [32]	✓	✓	✓	✓	✓	13

2.3 Machine learning

In this section, a brief overview of the field of machine learning is presented, followed by some details on more commonly used techniques that are utilised later on. In general, machine learning is a broad field where the aim is to model the structure of data or estimate relationships between observations.

Machine learning mainly falls into two broad categories, which are sometimes referred to as supervised learning and unsupervised learning. Given a set of observational data $\mathcal{D}$, it is often desired to find functional relationships between observations, or groups of observations. If the search for relationships in the set $\mathcal{D}$ is conducted using only $\mathcal{D}$, then the method is referred to as unsupervised learning. However, consider $\mathcal{D}$ indexed by i and a response set $\mathcal{G}$, also indexed by i. Then, inference on $\mathcal{D}$ trying to reproduce responses from $\mathcal{G}$ is referred to as supervised learning.

A distinction that can be made between machine learning methods are by the type of model they build. *Generative* models build estimates of the joint probability density over input variables of $\mathcal{D}$ (and $\mathcal{G}$, if supervised), but *discriminative* models only model the conditional density of outputs given the inputs. Discriminative models are only capable of predicting outputs given an input, but the model building process incorporates fewer assumptions due to not having to estimate joint densities over the input variables and responses. This often results in better performance with fewer training data. Since generative models find the joint distribution across the outputs and inputs, they can be used to find data points with low probability under the model, for novelty detection [33]. They may also be used to simulate new observations by sampling from the estimated generating distribution. However, with high-dimensional data (many features), in order to accurately estimate the joint density it is often necessary to have very

large amounts of observations, or otherwise use some simplifying assumptions. This can make the use of generative models infeasible in some cases, either due to lack of training data or assumption violations.

2.3.1 Supervised learning

Supervised learning can be broken down into two groups. If the responses $\mathcal{G}$ are real valued, i.e. $\mathcal{G}_i \in \mathbb{R}^p$, $\forall i$, then the task is referred to as *regression*. In this type of task there is assumed to be an unobserved function F which generates responses $\mathcal{G}$ from inputs $\mathcal{D}$, possibly corrupted linearly or non-linearly by some observation noise. The goal is to estimate an $\hat{F}$ which approximates F as well as possible given limited $\mathcal{D}$ and $\mathcal{G}$. However, if the responses in $\mathcal{G}$ are categorical (for example $\mathcal{G} \in \{1, 2, 3, ..., K\}$), then the inference task is referred to as *classification*. Rather than trying to learn a single function $\hat{F}$, a set of boundary determining functions, or decision functions are estimated in order to decide which category an observation belongs to.

Although too numerous to document here, some examples of popular supervised learning techniques include linear discriminant analysis, support vector machines (for classification or regression), artificial neural networks (for classification or regression), linear regression, and decision trees (for classification or regression). These techniques and many others, are described in detail in [33–36] among many other texts. Linear discriminant analysis and support vector machines are now discussed in detail, since these techniques are used throughout this thesis.

2.4 Linear discriminant analysis

Linear discriminant analysis (LDA) is a simple technique for finding a set of discriminant functions that differentiate between K categories. It often performs

as well as more sophisticated techniques under favourable conditions [34]. Linear decision boundaries are easy to estimate with relatively few parameters, and it scales well for many classes. A weight vector w_k and bias b_k for each class is estimated, and an observation vector, $\hat{x}$, to be classified is assigned to the class k which maximises

$$\arg\max_k f_k(\hat{x}) \ , \tag{2.2}$$

$$f_k(\hat{x}) = \sum_i \hat{x}_i w_{ki} + b_k \ . \tag{2.3}$$

In general, the K decision functions, f_k are given by

$$f_k(\hat{x}) = \log \pi_k - \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}\hat{x}^T\Sigma_k^{-1}\hat{x} - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k + \frac{1}{2}\hat{x}^T\Sigma_k^{-1}\mu_k + \frac{1}{2}\mu_k^T\Sigma_k^{-1}\hat{x} \ , \tag{2.4}$$

where $\log \pi_k$ is the prior probability of class k, Σ_k is the covariance estimate of class k, and μ_k is the estimated mean vector of class k. These quantities are estimated using training data. Under the assumption that Σ_k is the same for all k, these functions describe linear discriminant functions. The terms involving only Σ_k and $\hat{x}$ cancel across all k in (2.4). Then,

$$b_k = \log \pi_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k \ , \tag{2.5}$$

and, since $(\Sigma_k^{-1}\mu_k)^T = \mu_k^T\Sigma_k^{-1}$ due to symmetry of Σ_k^{-1},

$$w_k = \Sigma_k^{-1}\mu_k \ . \tag{2.6}$$

If Σ_k is not taken to be the same for all k, then instead the terms involving Σ_k and $\hat{x}$ do not cancel, and (2.4) provides quadratic discriminant functions. Decision rules formed using different covariances for each class is known as quadratic discriminant analysis (QDA), and may provide a better fit in some cases where

non-linear boundaries are not flexible enough. However, QDA suffers from having to fit many more model parameters than LDA (quadratic in the number of input dimensions), which is problematic when the amount of training data are small.

When not enough training data is provided to estimate Σ_k, i.e. number of predictors is larger than the number of training samples per category, or if some of the input predictors are highly correlated or nearly collinear, Σ_k will not be full rank. Then, performing inversion for (2.4) is not possible. Even if Σ_k is full rank, highly correlated predictors may lead to unsuitable selection of w_k, where it may appear to be not smooth, or have high variations. To address this, regularisation is imposed on Σ_k [37], where a penalty $\lambda\Omega$ is added to Σ_k. λ selects the amount of regularisation to apply, with $\lambda = 0$ giving no regularisation. Ω is a suitably designed penalty matrix. If $\Omega = \mathbf{I}$, then it is a penalty on the ℓ^2 norm of w_k. This type of penalisation is also known as *Tikhonov regularisation*. Discriminant analysis making use of Tikhonov regularisation or some other regularisation is referred to as penalised discriminant analysis.

These classification techniques can naturally handle multi-category problems. The main limitation of these methods is that for non-linearly separable classes, they are not flexible enough. Additionally, if the number of parameters in each observation is high (for current modern computing equipment, > 5000) the problem starts to become infeasible to solve, except in the case where there is much less data points than parameters; in this case, a computational shortcut [34] exists for LDA. LDA provides models that are easy to interpret, and due to often being a good performer in a classification setting, is a good first choice for classification tasks due to their simplicity.

2.5 Support vector machines for classification

Given a set of P training observations $(\underline{x}_1, \ldots, \underline{x}_P)$ of dimension l, with corresponding class labels $(y_1, \ldots, y_P)$, $y_p \in \{+1, -1\}$, a support vector machine (SVM) aims to find a decision surface which jointly maximizes the margin between the two classes and minimizes the misclassification error on the training set. When the classes are linearly separable, these surfaces are linear and have the form

$$f(\underline{\hat{x}}) = \sum_i \alpha_i y_i \langle \underline{\hat{x}}, \underline{x}_i \rangle + b = 0, \ \alpha_i \in \mathbb{R}^+ \ , \tag{2.7}$$

where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathbb{R}^l$, while the Lagrange multipliers α_i and the bias b are optimized by the training algorithm, and $\hat{x}$ is a test point to be classified as either 1 or -1.

In order to obtain non-linear decision boundaries, a mapping function $\phi(\underline{x})$ can be used,

$$f(\underline{\hat{x}}) = \sum_i \alpha_i y_i \langle \phi(\underline{\hat{x}}), \phi(\underline{x}_i) \rangle + b = 0, \ \alpha_i \in \mathbb{R}^+ \ . \tag{2.8}$$

This gives a linear boundary in ϕ, which corresponds with a non-linear surface in the original space. However, a certain class of functions exist, $K(\cdot, \cdot)$, known as *kernel functions*, which can be shown to be computing inner products in mapped spaces directly, without computing the mapping explicitly. Some examples are

$$K_r(\underline{a}, \underline{b}) = e^{-\gamma ||\underline{a} - \underline{b}||^2}, \qquad \gamma \in \mathbb{R}^+ \tag{2.9}$$

$$K_p(\underline{a}, \underline{b}) = (k + c\langle \underline{a}, \underline{b} \rangle)^d, \qquad k, c \in \mathbb{R}, d \in \mathbb{Z} \ . \tag{2.10}$$

In the spaces for which these inner products are computed, the data could potentially be linearly separable. Analogously to the linearly separable case, the

decision surface is constructed according to

$$f(\underline{\hat{x}}) = \sum_i \alpha_i y_i K(\underline{\hat{x}}, \underline{x}_i) + b = 0 \ , \tag{2.11}$$

and the class label of a test vector $\underline{\hat{x}}$ is predicted to be the sign of the score function evaluated at $\underline{\hat{x}}$,

$$C(\underline{\hat{x}}) = \text{sgn}(f(\underline{\hat{x}})) \ . \tag{2.12}$$

The main benefit of the SVM method is the introduction of slack variables in its formulation, allowing for training misclassification without skewing the orientation of the decision boundary in a way that could potentially reduce generalisation performance.

2.5.1 Optimising SVM parameters

SVMs involve some free parameters that need to be optimised in order to achieve good classification performance on unseen examples. For SVMs using the radial basis function (RBF) kernel (2.9) or polynomial kernel (2.10), these parameters are:

$\mathcal{C}$: Trade-off between margin width and misclassified points in the training data. Large values mean the SVM tries to fit the training data more exactly, admitting fewer training errors, at the expense of reducing the margin from the boundary to points of each category. This value also acts as the upper bound of the Lagrange multipliers α_i.

γ: RBF kernel parameter which controls the width of the Gaussian function. Small values of γ lead to decreasing flexibility of the decision boundary, while large values of γ allow a more flexible boundary.

k: Polynomial kernel parameter, often selected to be either 0 or 1, effectively controlling the presence of cross-terms in the polynomial expansion.

c: Polynomial kernel parameter controlling fit flexibility by weighting the contribution of inner products between points. Smaller c will result in less flexible fits, and larger c will result in more flexible fits.

d: Polynomial kernel parameter controlling fit flexibility by parameterising the number of turning/saddle points expressed at the decision boundary

It is not immediately intuitively clear which values for these parameters would be the best for any given problem, despite the fact that they are "user-selectable" parameters (often referred to as *hyper parameters* to avoid confusion with the parameters estimated within training procedures, such as μ and Σ in LDA/QDA or α_i and b in SVMs).

These parameters can be tuned by means of searching over a grid of the hyper-parameters. Many SVMs are trained, each with a different value for each hyper-parameter within a given range. Using separate validation data, the generalisation ability is estimated, and the position on the hyper-parameter grid which minimises a chosen metric on test error is used to select the final hyper-parameters, for training an extended dataset.

2.5.2 Multiclass classification using SVMs

It is clear from the definition given in 2.5 that SVMs are binary only classifiers. This poses a problem in the case of multi-category tasks. A few approaches to using SVMs for multicategory tasks are outlined, and the method of choice for this thesis is elaborated in some more detail.

There are two possible ways to approach the problem. Either the SVM formulation can be updated to naturally handle multi-class classification, or some binarisation techniques can be adopted for the purpose of breaking down the overall problem into many smaller binary problems. Treatments for multi-class SVMs are given in [38, 39]. However, although optimal in the sense that the

margins for all categories are jointly optimised, they can suffer from considerably slower training times [40], in part due to the increasing amount of kernel evaluations with the number of categories. On the other hand, whilst not treating the problem optimally, binarisation techniques are popular for simplicity and the ability to train (and test) the classifiers in parallel.

Strategies for creating multi-class classifiers from binary SVMs involve partitioning the data by their base categories into other groups, or excluding some categories. A number of SVMs are trained with the original category labels mapped to $\{1, -1\}$, and then some aggregation is applied to the outputs in order to determine the final category. Popular techniques for class binarisation include the *one vs one* and *one vs all* schemes. With one vs one, each category is trained against every other category, which for the K-category problem requires training $\frac{K^2-K}{2}$ different classifiers. Only K classifiers need to be trained with one vs all approach, where one category is isolated, and the remaining categories are treated as a single category. Then, with either approach, a variety of aggregation methods can be utilised for combining the classifier outputs into a single decision. Majority voting is often favoured for one vs one, whilst most confident vote is favoured for one vs all. In addition, many more aggregation strategies are available for one vs one and one vs all binarisation, which are covered in [41].

Other approaches to binarisation strategies use directed acyclic graphs [42], maximum margin trees [43], and error correcting output codes [44, 45]. It should be noted, however, that for directed acyclic graphs and margin trees, both would likely result in the same solution for the three category problem, like the one in this thesis. However, in order to make use of error correcting codes method for three categories, both one vs one and one vs all binarisation must be used. This is actually advantageous as it introduces classifier diversity that can potentially improve accuracy. A brief treatment of the error correcting code method in the specific context of the three class problem is given next.

2.5.3 Multiclass SVMs using error correcting codes

Binary SVM classifiers are combined via predefined error correcting output code methods [44,45]. To achieve this, N binary classifiers are trained to distinguish between M classes using a coding matrix $\mathbf{W}_{M\times N}$, with elements $\mathbf{W}_{mn} \in \{0, 1, -1\}$. SVM n is trained only using data from classes for m such that $\mathbf{W}_{mn} \neq 0$, with $\mathbf{W}_{mn}$ as the class label. Then, the class assignment rule is given by

$$C(\underline{x}) = \arg\min_{m} \sum_{n=1}^{N} \chi(\mathbf{W}_{mn} f_n(\underline{x})) \ , \tag{2.13}$$

where $f_n(\underline{x})$ is the output of the n^{th} SVM and χ is some loss function.

The error-correcting capability of a given code is commensurate with the minimum Hamming distance between pairs of rows of the coding matrix; if this minimum distance is δ, then the decoding process will be able to correct any $\lfloor \frac{\delta-1}{2} \rfloor$ errors [44]. It is for this reason that an error correcting code with only 3 classifiers is insufficient in the 3 class case, since the minimum Hamming distance is 2, allowing no error corrections. Therefore, consider all one vs one and all one vs all binary classifiers, which makes a total of six binary classifiers. In the case of three classes, this exhausts all possible binary classifiers, with pairwise distances between rows as high as 5. The corresponding coding matrix in this case thus has the form

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & -1 & 1 & 1 & 0 \\ 1 & -1 & 1 & 0 & -1 & 1 \\ -1 & 1 & 1 & -1 & 0 & -1 \end{bmatrix} . \tag{2.14}$$

A number of choices for the loss function χ exist, including

$$\chi(z) = \begin{cases} \max(1-z, 0), & \text{hinge loss} \\ \frac{1}{2} - \frac{1}{2}\text{sgn}(z), & \text{Hamming loss} \\ e^{-z}, & \text{exponential loss} \\ -z, & \text{linear loss} \end{cases} \tag{2.15}$$

2.5.4 SVM training with unbalanced categories

For the NVR vs VT vs VF problem, it is to be expected that most data that will be tested will be NVR. This may be less true in an AED setting, but will be true of many other settings, such as in AICDs. In fact, the curated data used for developing and assessing algorithms which is discussed in 2.8 is mostly NVR, with a small amount of VF and even less VT.

Consider again, the SVM equation defining the separating hyperplane (2.11)

$$f(\underline{\hat{x}}) = \sum_i \alpha_i y_i \langle \phi(\underline{\hat{x}}), \phi(\underline{x}_i) \rangle + b = 0, \ \alpha_i \in \mathbb{R}^+ \ .$$

The training procedure essentially searches for optimal values of α_i, which is solved as an optimisation problem, subject to some constraints. In particular, one constraint is of interest;

$$\alpha_i \leq \mathcal{C} \ . \tag{2.16}$$

Inspecting this in the context of (2.11) and unbalanced training data, it is clear that a learned hyperplane may favour the majority class by virtue of more support points ($\alpha_i \geq 0$) coming from the majority class. This effect is particularly profound in the non-separable data case, since $\alpha_i = \mathcal{C}$ for misclassified training points.

There are a variety of techniques for dealing with unbalanced categories in

SVM training [46], which include undersampling the majority class, synthesising new points for the minority class, or an improved optimisation algorithm for training SVMs with different values of $\mathcal{C}$ per category, usually weighted by the class imbalance ratio. Generative models may be used to derive points for synthetic upsampling of minority classes, but this approach makes little sense, since the minority class has fewer points, unlikely enough to build a reasonable generative model to begin with. Additionally, upsampling would cause an increase in SVM training times. On the other hand, different costs allows the use of the full data set without the risk of learning a hyperplane biased by the majority category, but requires implementation support. Undersampling the majority category can result in substantially faster training times without any implementation requirements, at the cost of some accuracy. However, when the amount of even the minority class is not small, it is conceivable that a reasonable hyperplane will still be learned. Undersampling is the method employed in this thesis, due to training times required, and the fact that GPU software used for accelerating SVM computations [47] does not support different costs per category.

2.6 Model selection and error estimation

So far, a couple of methods for learning a model to separate labelled sets of observations have been discussed. However, it is necessary to properly evaluate the ability of a learned model to generalise with unseen observations. There are a few non-parametric methods for estimating the generalisation ability.

2.6.1 Data hold out for generalisation estimates

The simplest of these methods is data hold out. Essentially, in a data rich situation, the total data is partitioned into two (or three) subsets. One subset is used for developing a model. If model development requires parameter tuning,

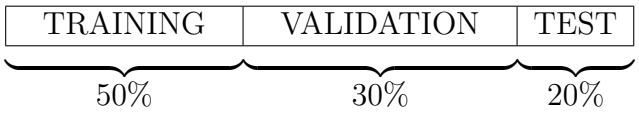


Figure 2.1: Data hold out example splits. A validation set is only required for methods where tunable parameters need to be selected

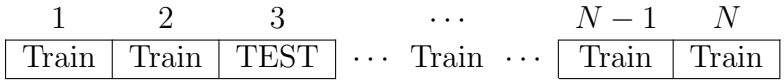


Figure 2.2: Data is partitioned into N roughly equally sized sets after randomisation. It is important to ensure that relative class densities are similar to those of the overall data set. Each set is used as a test set once

such as with SVMs, the second set is used for assessing accuracy for each trained model. Finally, when tunables have been selected, the model is built and the third set is used for assessment of accuracy. A typical split scheme for data hold out is shown in Figure 2.1.

2.6.2 Cross-validation for generalisation estimates

In practice, data hold out is rarely used, often because there is not sufficient data to split. Even in cases where there is, it may not be feasible to build models using the entirety of the data. Thus, alternative techniques may be employed. Cross validation splits the data into N roughly equally sized partitions, and holds one partition back while using the remainder for testing, with care being taken to preserve class ratios. This is repeated with each partition being held back once for testing. Then, an average metric and standard deviation of those metrics can be computed. Depending on how much training data is required for good generalisation ability, more or less folds may be used. In the extreme case, a single data point is held out and the rest is used for training, which is known as leave one out cross validation. Such an estimate of generalisation ability is unbiased, however N-fold tends to underestimate generalisation error. Cross validation may also be sensitive to the initial data split. The general N-fold cross validation scheme is shown in Figure 2.2.

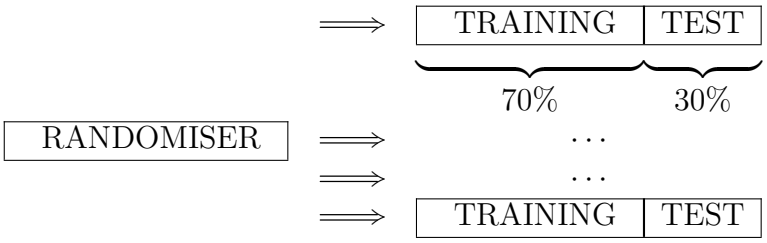


Figure 2.3: Bootstrap resampling is performed by randomising the data, which is then split into portions to be used for training (and perhaps development) and testing. This is repeated several times with different randomisations each time

2.6.3 Bootstrap resampling for generalisation estimates

One of the issues with cross validation is that it may be sensitive to the way the data is initially split. Additionally, no independence of the test set size is possible, due to the size of each set being $\frac{1}{N}$ of the data set. Bootstrap resampling can address these problems because each evaluation of the model building procedure is performed using independent draws of test and training sets. This also allows for appropriate sizing of the training and testing sets independently of the number of the draws. As originally proposed, bootstrap resampling evaluates accuracy metrics using the entire dataset, which would be a biased estimator since a large proportion of each evaluation would actually be training re-substitution. A better way to use bootstrap resampling is to use only the non-training portion for evaluation. This can increase the computational requirement however, since there should be enough draws in order to have a reasonable probability that all data points appear roughly the same number of times in the test sets. Figure 2.3 shows the bootstrap resampling scheme.

2.6.4 Metrics for classification

Given a set of category labels $\mathcal{L}$, a set of categories S, and a label assigning function, the following quantities can be defined, for each $s \in S$;

1. TP_s as the number of category s correctly assigned to s,

2. FP_s as the number of other categories incorrectly assigned to s,

3. FN_s as the number of category s incorrectly assigned to other categories, and

4. TN_s as the number of other categories not assigned as category s.

With these quantities, an accuracy metric can be defined as

$$Acc = \frac{\sum_s TP_s}{|\mathcal{L}|} \ . \tag{2.17}$$

Usage of this metric suffers from the problem that if the amounts of categories are unbalanced, then a label assigning function which assigns all labels to the majority category can have a misleadingly high accuracy. If values for s are restricted to $\{pos, neg\}$, then the traditional definitions of sensitivity and specificity are

$$\text{sensitivity} = \frac{TP}{TP + FN} \ , \tag{2.18}$$

$$\text{specificity} = \frac{TN}{TN + FP} \ . \tag{2.19}$$

Rewriting these, it becomes clear that specificity is nothing more than sensitivity of neg category

$$\text{sensitivity} = \frac{TP_{pos}}{TP_{pos} + FN_{pos}} \tag{2.20}$$

$$\text{specificity} = \frac{TP_{neg}}{TP_{neg} + FN_{neg}} \ , \tag{2.21}$$

and therefore, a general expression for sensitivity sn (or recall in machine learning literature) is given as

$$sn_s = \frac{TP_s}{TP_s + FN_s}, \ s \in S \ , \tag{2.22}$$

The same generalisation can be applied to positive predictive value (precision

in machine learning literature) and negative predictive value, thus generalised positive predictive value *pp* is given as

$$pp_s = \frac{TP_s}{TP_s + FP_s} \; s \in S \; . \tag{2.23}$$

Now, balanced accuracy, or average sensitivity, can be defined as

$$Acc_{bal} = \frac{1}{|S|} \sum_{s \in S} sn_s \; . \tag{2.24}$$

Other per-class metrics which can be derived from the generalised sensitivity and generalised positive predictive value are the per category G-score;

$$G_s = \sqrt{sn_s \cdot pp_s} \; , \tag{2.25}$$

and per category F_β-score;

$$F_{\beta_s} = (1 + \beta^2) \; \frac{pp_s \cdot sn_s}{(\beta^2 \cdot pp_s) + sn_s} \; , \tag{2.26}$$

with β being used to weight sensitivity or positive predictivity higher ($\beta = 1$ for equal weighting). Each of these can similarly be averaged across all categories, as with (2.24).

For some binary learning tasks a receiver operating characteristic (ROC) analysis is conducted in order to understand how the false alarm rate varies with the true alarm rate. The ROC is produced by varying some parameter of the learned model to introduce a higher true alarm or higher false alarm rate. A complementary metric to the ROC is the area under the curve (AUC), in order to produce a single value from the curve which provides an informative single metric. An AUC at 0.5 is a classifier performing no discrimination, while as the AUC value approaches 1, the classifier is understood to be better at separating

false and negative instances.

2.7 Unsupervised learning

All of the considerations so far have been with respect to supervised learning, in particular classification, where some ground truths corresponding with observations are known. The main limitation with these methods is that such labels are not always readily available, often requiring many domain experts to spend time producing the ground truths. On the other hand, unlabelled data is often high in availability. In this case, if a generating function for such unlabelled data can be estimated, useful insights might be obtainable.

Examples of popular unsupervised learning techniques include; principal component analysis for subspace or feature extraction; independent component analysis for source separation, feature extraction or dimension reduction; K-means, K-mediods and Gaussian mixture models for clustering; 1-class SVMs and Parzen windows for density estimation; and multidimensional scaling and unsupervised neural networks for manifold learning. Some details for principal component analysis are provided, since this method is utilised later on in the thesis.

2.7.1 Principal component analysis

Often, the variance of a data set is not spread uniformly along its input dimensions. If linear combinations of input dimensions are responsible for considerable variation of the data, this structure can be captured. Principal component analysis (PCA) is a technique to find a set of basis vectors and their corresponding responsibilities for explaining the variance of the observed data. Reconstruction of observations is simply the sum of each basis function multiplied by the corresponding basis vector weight for that observation. In this way, progressively better approximations, on average, of the original observations can be formed by

inclusion of more of the weighted basis functions in the order of their rankings, up to a tolerable amount of reconstruction error.

This is the basis for structured *dimension reduction*. To compute the basis vectors, assume data matrix $\mathbf{D}$, with N rows of input dimension P. First, the mean along each input dimension is subtracted

$$\hat{\mathbf{D}} = \mathbf{D} - \boldsymbol{\mu}, \tag{2.27}$$
$$\boldsymbol{\mu}_{kj} = \sum_{\forall i} \mathbf{D}_{ij},\ j \in \{1, \ldots, P\},\ \forall k\ .$$

Then, the basis functions can be computed by solving the eigenvalue equation of the covariance matrix,

$$\frac{\hat{\mathbf{D}}^T \hat{\mathbf{D}}}{N} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T\ . \tag{2.28}$$

The columns of $\mathbf{U}$ are the basis functions, and the diagonal of $\boldsymbol{\Lambda}$ are the eigenvalues indicating the amount of variance contained in the direction spanned by each eigenvector. It is then possible to pick a set of eigenvectors, $\hat{\mathbf{U}}$, and eigenvalues, $\hat{\boldsymbol{\Lambda}}$ that contain, for example, the top 90% of the variance using $\boldsymbol{\Lambda}$. Finally, the data can be presented in the reduced subspace by simple matrix multiplication with the basis functions, $\mathbf{B} = \hat{\mathbf{U}}$,

$$\mathbf{D}' = \mathbf{D}\mathbf{B}^T\ , \tag{2.29}$$

or, if the data is desired to be standardised across all output dimensions taking the basis functions as $\mathbf{B} = \hat{\mathbf{U}}\hat{\boldsymbol{\Lambda}}^{-\frac{1}{2}}$. For many statistical learning techniques, standardising the input is often useful as methods may be dominated by input variables that simply have a larger range, but not necessarily higher significance, for example, SVMs using non-linear kernel functions.

For any new data $\mathbf{D}_{\text{new}}$, these may also be projected onto the same co-ordinate system using (2.29) by simply replacing $\mathbf{D}$ with $\mathbf{D}_{\text{new}}$.

2.8 ECG databases

In order to test existing and proposed procedures for their ability to differentiate between NVR, VT and VF, a database with labelled rhythms is required. Physiobank [48] maintains a large, publicly available, online repository of various physiological signals, including annotated ECG signals. Four databases from Physiobank were chosen, these were the European ST-T Database (EDB) [49], the Creighton University Ventricular Tachyarrhythmia Database (CUDB) [50], the MIT-BIH Arrhythmia Database (MITDB) [51] and the MIT-BIH Malignant Ventricular Arrhythmia Database (VFDB) [52]. All of these have been used at some point in previous studies, apart from the EDB, which is included to augment the dataset with examples of VT. Another database commonly used in previous studies is the American Heart Association Database (AHADB) [53]. It is unclear in various instances whether the AHADB used is the older release or the extended version, however, for this thesis the extended version was obtained. Although not freely available, the AHADB can be acquired for a reasonable cost. This database is frequently used in ECG studies and has been available for considerable time, so despite being non-free to obtain, the AHADB is considered to fulfil "publicly available data" requirements.

These databases vary in several parameters, including; sampling rate; quality and type of annotations; number and position of leads; and record length. A variety of these properties are shown and described for the databases, and then data preprocessing is described.

Table 2.2: Properties of the chosen ECG databases, including record length, sampling frequencies, number of ECG channels and the type of annotations present

Database	Sampling Frequency (Hz)	Number of Records	Record Length (mins)	Channels	Annotations
Extended AHADB	250	154	35	2	Rhythm and beats
VFDB	250	22	35	2	Rhythm only
EDB	250	90	120	2	Rhythm and beats
CUDB	250	35	8.5	1	Rhythm only
MITDB	360	48	30	2	Rhythm and beats

Table 2.3: Non exhaustive list of the main rhythms present in each of the databases

Database	Rhythms present
Extended AHADB	normal rhythms, paced rhythms, bigeminy, ventricular flutter and VF (undifferentiated)
VFDB	normal rhythms, paced rhythms, bigeminy, AF, supraventricular tachyarrhythmias, escape rhythms, VT, ventricular flutter, VF
EDB	normal rhythms, AF, bigeminy, trigeminy, supraventricular tachyarrhythmias, VT
CUDB	normal rhythms (anything not AF, VT or VF is categorised as this), VT, ventricular flutter and VF (undifferentiated), VF (differentiated)
MITDB	normal rhythms, paced rhythms, pre excitations (Wolff-Parkinson-White), supraventricular tachyarrhythmias, AF, differentiated ventricular flutter, VT, ventricular flutter and VF (undifferentiated)

2.8.1 Database statistics and rhythm labelling

Each database is composed of a number of possibly multichannel records, which in most cases (with a few exceptions[1]) belong to unique patients. In cases where the documentation does not provide this information, or it is unavailable, it is assumed that the data comes from unique patients. All databases apart from the CUDB are multichannel, however the lead positioning from patient to patient is not documented in some cases, and varies considerably in others. Thus, the data that is used, is simply taken from the first provided channel for each record.

Table 2.2 shows some higher level information about each of the databases, including; the ECG sampling frequency, number of records per database, record length, number of ECG channels, and what form of ECG annotations (rhythm or beats) are present.

[1]This is documented by the MITDB `http://www.physionet.org/physiobank/database/html/mitdbdir/intro.htm#selection`, with 48 records from 47 patients, and the EDB `http://www.physionet.org/physiobank/database/edb/`, with 90 records from 79 patients

The databases contain many types of beats and rhythms. Table 2.3 describes the majority of rhythms present in each database. Of particular note, is that the labelling conventions in some cases appear to group ventricular flutter and VF together. For some databases, distinct labelling for ventricular flutter and VF exists, and in some cases, distinct ventricular flutter is labelled, along with a single category for undifferentiated ventricular flutter and VF. Since ventricular flutter is a poorly defined concept, and not acknowledged in the Lambeth Conventions [10, 11], labels for undifferentiated ventricular flutter and VF are treated as VF.

Ventricular bigeminy and ventricular trigeminy are rhythms involving premature ventricular contractions, but the isoelectric level is still detectable. This is in contrast with VT, which is essentially a run of four or more premature ventricular contractions where no isoelectric level is detectable. Therefore, despite some similarities in morphology between VT, and ventricular bigeminy/trigeminy, the latter, and premature ventricular contractions are categorised as NVRs in this work. It is worth noting that the rhythm annotations for VT do not differentiate between monomorphic or polymorphic VT, which might be problematic in a clinical setting if different treatment is indicated based on whether a VT is polymorphic or monomorphic.

The extended AHADB has records in long and short form. The long records are each 3 hours long, with annotations on the last 30 minutes. The short form records are comprised of the last 35 minutes, therefore the first 5 minutes of the short records are not annotated. These unannotated sections are assumed as being NVRs; a cursory check did not suggest that any of the unannotated sections were VT or VF.

As mentioned previously, ventricular flutter is a somewhat contentious rhythm label, but a small amount of data is labelled as this rhythm. Since the relative amount of differentiated ventricular flutter in the databases is minimal, and it is not clear whether to mark them all as VT or VF, the only remaining option is to

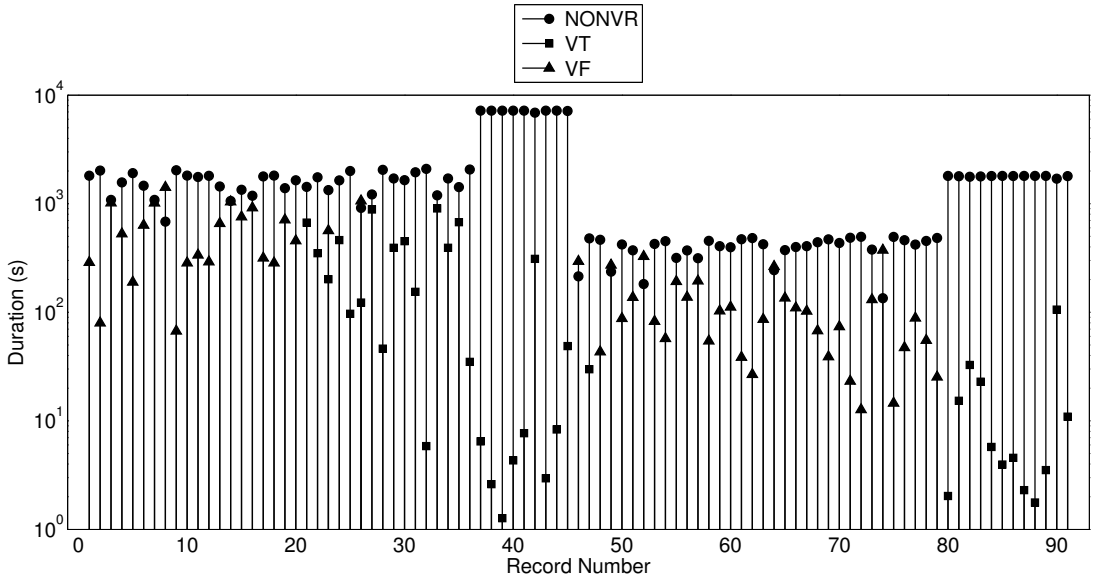


Figure 2.4: Distribution of rhythms present in each patient record. Most records contain only labelled VT or VF, but not both. The amounts shown are durations in seconds, per rhythm per record, shown on a logarithmic scale

exclude records containing this rhythm. All rhythms that are not labelled as VT, VF or ventricular flutter, are relabelled as NVR (including some small amounts of asystole which are present post-shock in the databases). Then, records are selected which contain either VT or VF (including undifferentiated ventricular flutter and VF), and records containing any explicitly labelled ventricular flutter are excluded. There are 98 records containing VT or VF, 7 of which contain explicitly labelled ventricular flutter. Thus a total of 91 records are used. Despite some records documented to be coming from the same patient, after the record selection procedure, no patient appears twice in the resulting dataset, assuming that records belong to unique patients in the other databases.

Figure 2.4 shows the duration of each category present in each of the selected 91 records, and Figure 2.5 shows the total duration of each category across the entire set of 91 records to be used from the databases.

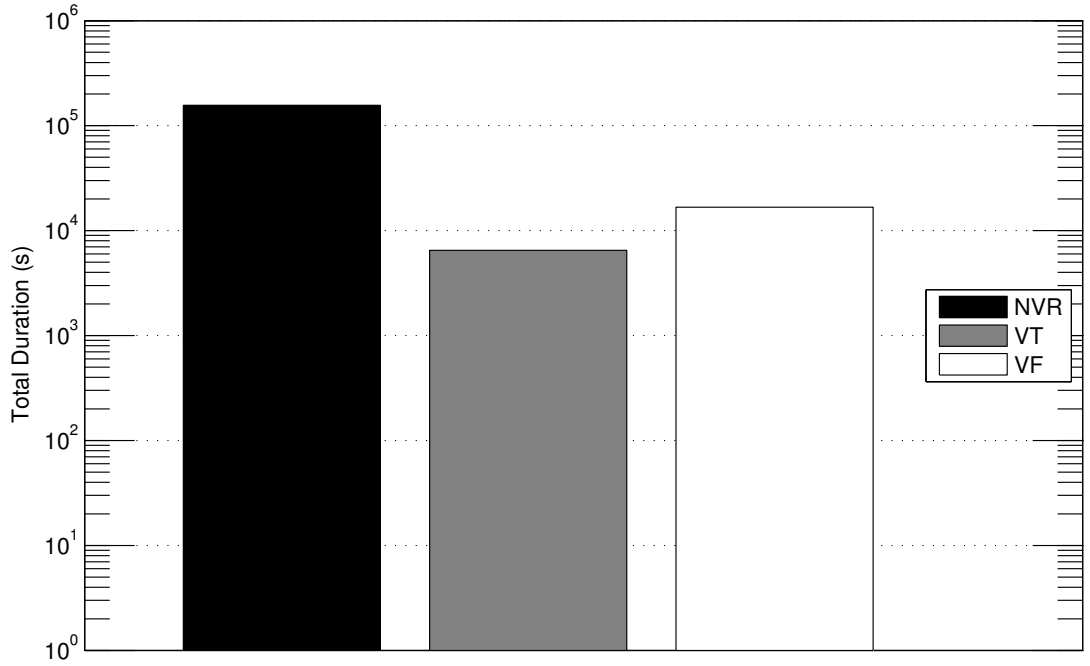


Figure 2.5: Total amount of each of NVR, VT and VF across all the patient records, shown in logarithmic scale. NVR is almost 10x more present than other rhythms

2.8.2 Data preprocessing

Prior to window segmentation and feature extraction, the records should be preprocessed to be consistent among each other in parameters, and also to prevent the presence of some database specific features that might either hinder or artificially help classification accuracy (for example, VT and VF are largely not present simultaneously according to the labels). Some low frequency artefacts due to mechanical properties of the tape recorders used to originally obtain the ECG recording may be present. Also, the databases are not all sampled at the same rate, and some are low pass filtered as low as 70 Hz. Additionally, another study, on the frequencies present during VF used a sampling rate of just 100 Hz [54].

Therefore, the preprocessing that is described is based on a mixture of motivations, from a physiological standpoint, and to ensure no bias is present that may influence the results.

Since the lowest value for low pass filtering at acquisition time is known as

70 Hz, all records should be low pass filtered at 70 Hz or lower. It is considered that most of the relevant information is contained in the 40 Hz baseband [55] and previous studies obtain experimental results using a second order low pass Butterworth filter with a 30 Hz cutoff [3, 5, 16, 21, 26]. However, such a filter has a non-linear phase characteristic that may be undesirable, and higher levels of attenuation would only be obtained at higher frequencies.

Since the databases need to be re-sampled due to difference in sampling rate, and the different sampling rates do not have common factors, upsampling must be performed prior to decimation. It turns out that resampling to 100 Hz sampling rate requires the lowest upsampling factor for all databases involved. Therefore a finite impulse response filter is preferred, as there are computational shortcuts for this class of filters, and the upsampling interpolation and lowpass filtering can be combined. Additionally, this class of filters have the advantages of constant group delay and no phase distortion. A windowed sinc function is used to design a finite impulse response filter with a 49 Hz cutoff. This maximises flatness at the passband, (unlike with a second order Butterworth filter, where the slow roll off region can be considered to be part of the passband still), with a sharp transition. This allows the records to be downsampled to 100 Hz.

Finally, a highpass finite impulse response filter with cutoff 0.4 Hz is designed to remove baseline wandering [55] and other low frequency artefacts that may be due to patient breathing or movements. Then, each record is normalised such that it's sum of squares is equal to the length of the record, in order to compensate for acquisition devices with different gains.

2.9 Summary and conclusions

Methods for rhythm diagnostics in the ECG were explored, varying from methods considering just NVR vs arrhythmias, to methods considering NVR vs VT vs

VF. A selection of methods categorised by their operating domain were briefly described, and the mode of obtaining diagnoses was discussed. It was found that in many cases, formal methods from machine learning were not utilised, and often ad-hoc decision methods were developed. Additionally, the limitations and experimental problems of some of these methods were enumerated.

Then, the field of machine learning was introduced briefly, and specific techniques were discussed. Particular attention was paid to the description of SVMs, which are used extensively in this thesis. This included a discussion of the issues around multiclass classification, hyperparameter optimisation and SVM training in the presence of unbalanced categories. A brief overview of unsupervised learning was provided, as well as a description of the PCA method as a dimension reduction technique. Methods for assessing classification accuracy were also described, including a variety of possible metrics, and methods for obtaining estimates of generalisation ability. These considerations are key for the design of an appropriate experimental framework in the thesis.

Finally, the data to be used throughout experimental investigations in this thesis was discussed. Five databases were included, due to their public availability and ubiquity of use in ECG diagnostics research. Important parameters of the database are described, such as the sampling rate, number of leads and types of rhythms present. Exclusion criteria for records was explained, and for the records chosen to be used, some statistics on rhythm prevalence were provided, per ECG record and aggregated.

Chapter 3

Preliminary investigations

Given the requirements of the task, differentiate between NVR, VT and VF, it is useful to first conduct some exploratory experiments, and to develop previous methods for comparison purposes. A commonly made recommendation to approaching a new pattern recognition task is to try the easy and computationally cheap methods such as LDA first, which can often given adequate performance.

Therefore, in this chapter, feature representations are developed, and some basic classification schemes are evaluated. Results from these investigations are then built upon in subsequent chapters.

A preliminary form of the work in this chapter is elaborated [56, 57] using a methodology for assessing accuracy which is not to be considered as valid as the methodology presented in this thesis.

3.1 Introduction

Previous studies performing analysis on ECG rhythms usually take a windowing approach to diagnosis, i.e. given a segment $\underline{x}$ of discretised ECG signal,

$$\underline{x} = \{x[n], n_1 \leq n \leq n_2\} \ , \tag{3.1}$$

find the class $C(\underline{x})$, usually of the form

$$C(\underline{x}) = f(T(\underline{x})) \ , \tag{3.2}$$

where T is some transformation function whose output is a vector of features, and f is some decision function. There are some examples of algorithms which operate differently, e.g. [4], but these type of approaches are uncommon. Of particular interest, are the functions T developed in prior art which extract and output features from the ECG. A good choice of T is essential in order for the decision function f to be able to perform well.

In this chapter, a selection of different T are presented and evaluated in their combined discriminative capability between VT and VF. Using features derived from non-overlapping segments of ECG for different segment lengths, estimates of classification accuracy using LDA, QDA and SVM classifiers are made using bootstrap resampling.

Two recent studies aimed to select the most relevant features from the literature for NVR vs arrhythmia and non-VF vs VF tasks; this second task is closer to the goal of assessing VT and VF discrimination. Feature recommendations and observation length recommendation were made by these studies, which guides the selection of features used in this thesis for comparative purposes. The classification task throughout will be three-way discrimination between NVR, VT and

VF. While this is not strictly an assessment of VT vs VF discriminability, it is important to assess a realistic scenario.

3.2 Methods

3.2.1 Transformation features for comparison purposes

Recently, some studies were conducted with the aim of identifying which transformations from previous works are the most useful for arrhythmia detection [31,32]. Some selection procedures are used to rank combinations of features from the literature for arrhythmia vs NVR detection [31, 32], or non-VF (NVR and VT) vs VF detection [32]. There are some common features between the two studies, but since [31] only assesses NVR vs arrhythmia, less features from this study are selected. The features used for comparison purposes in this thesis will be described. All of the described features are easily normalised to be in the range $[0, 1]$.

1. **VF filter leakage** [58] is obtained by a processing technique which corresponds with applying an adaptive bandstop filter centered at the mean frequency of the considered window $\underline{x}$ and measuring the residual energy. If the mean period T of the ECG window of length m is computed as

$$T = \frac{\pi \sum_{i=1}^{m} |\underline{x}_i|}{\sum_{i=2}^{m} |\underline{x}_i - \underline{x}_{i-1}|} , \tag{3.3}$$

then the leakage is computed as

$$\text{Leakage} = \frac{\sum_{i=1+T}^{m} \left|\underline{x}_i + \underline{x}_{i-T}\right|}{\sum_{i=1+T}^{m} |\underline{x}_i| + \left|\underline{x}_{i-T}\right|} \ . \tag{3.4}$$

2. **Count 2** [3] is obtained by applying a bandpass filter to $x[n]$ and then computing some statistics on the output. The filter was originally implemented as a recursive integer filter, with intended central frequency of 14.6 Hz and 3dB bandwidth from 13 Hz to 16.5 Hz. For a 250 Hz sampled signal this filter is given as

$$8y[n] = 14y[n-1] - 7y[n-2] + \frac{(x[n] - x[n-2])}{2} \ . \tag{3.5}$$

This is reimplemented by designing an equivalent filter with the same passband and stopband characteristics for 100 Hz sampling, which is used in place of the integer filter. Then, the **Count 2** statistic is computed as the number of samples fulfilling the criterion

$$\text{mean}(|y[n]|) \leq |y[n]| \leq \max(|y[n]| \ , \tag{3.6}$$

where $y[n]$ is the filtered output.

3. **Threshold crossing sample count** [15] is an improvement to the threshold crossing interval [13] transformation, obtained by counting number of samples above the absolute value of an adaptive threshold. The segment $\underline{x}$ is multiplied with a Tukey window where the centre half of the window is constant, and the remainder tapers to 0. Then, the maximum absolute value of $\underline{x}$ is found, and multiplied by 0.2 to obtain the threshold τ. Finally,

the **Threshold crossing sample count** is the number of samples fulfilling the criterion $|\underline{x}| \geq \tau$, i.e.

$$\sum_i \tau\left(\underline{x}_i\right), \text{ where } \tau\left(\underline{x}_i\right) = \begin{cases} \tau(\underline{x}_i) = 1, & |\underline{x}_i| \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

4. The **sample entropy** of $\underline{x}_i$ [25] is used for quantifying the amount of self similarity and is used directly as a feature. For a sequence of length N, an embedding dimension m is selected, usually 2. Then, all possible subsequences of $x[n]$ of length $N - m - 1$ are found, and pairwise distances between all of these subsequences are computed, usually using Chebychev distance, but any distance metric is sufficient. Then, the number of distances with a value above some threshold r is computed as A. The value B is computed in the same fashion, instead using subsequences of length $N - M$. Finally, the **sample entropy** of $x[n]$ is given as

$$SampEn(x[n]) = -\log \frac{A}{B} \ . \tag{3.8}$$

The threshold for the distances, r, is usually computed as $0.2 \times \sigma$, where σ is the standard deviation estimated given a sequence $\underline{x}_i$.

5. **Spectral parameter** m [19]. To compute this parameter, first the discrete Fourier transform of $x[n]$ is found and F, the frequency with the largest amplitude between 0.5 Hz and 9 Hz, is identified. Then m is computed as

$$m = \frac{\sum_i A_i f_i}{F \sum_i A_i} \ . \tag{3.9}$$

where f_i is the i^{th} frequency, and A_i is the absolute value of the discrete

Fourier transform at f_i.

6. **Spectral parameter** $A2$ [19]. Using F, A and f as defined above,

$$A2 = \sum_i A_i, i \in 0.7F \leq f_i \leq 1.4F \ . \tag{3.10}$$

7. **PST** [26] phase space is formed by taking $x[n]$ and its shift by 0.5 s. This phase space is quantised into a 40×40 grid and the number of the unique value-pairs are counted.

8. **PSH** [27] phase space is formed by taking $x[n]$ and its Hilbert transform. As above, the phase space is quantised, and the number of unique value-pairs are counted.

When performing comparative analysis, two groups of these features are formed and named. Two of the above are recommended [31], that is VF filter leakage and Count 2 (1 and 2 from above) for NVR vs arrhythmia classification. In this thesis, this combination of input features is referred to as *Heur2*. The remaining 6 features are the highest ranked features by the feature selection method and AUC analysis [32] for the non-VF vs VF task. The VF filter leakage feature is also among those features highly ranked in this study. The collection of all 8 features described is referred to as *Heur8*.

3.2.2 High dimensional transformation features

As noted in Section 2.2.4, features in prior studies extracted from ECG segments are relatively low dimensional. In this thesis, the benefit of high dimensional representation spaces are explored. In preliminary studies [56, 57], raw ECG samples, their Fourier magnitude spectra, and principal components of their Fourier magnitude spectra were tested for their classification performance. For

the more difficult binary classification task of VT vs VF, raw ECG samples were not able to perform better than the level of guessing. Therefore these features are not explored and presented in this thesis. The Fourier magnitude spectra and their principal components representation spaces are described next.

For a given segment of ECG of length N, the discrete Fourier transform is a complex valued transform which is computed as

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-i2\pi kn/N}, k \in \{0, \ldots, N-1\} \ . \tag{3.11}$$

In general, $X[k]$ is complex valued, however, since the input segment of ECG is real valued, $X[k]$ has the property of being complex-conjugate symmetric. Since the features would be formed from $|X[k]|$, the second half of samples of $|X[k]|$ can be omitted, since for a given complex number $\mathcal{Z}$, $|\mathcal{Z}^*| = |\mathcal{Z}|$, where * denotes complex conjugation. In addition, since the 0 frequency or DC component is not expected to be informative, $X[0]$ is also omitted.

The use of $|X[k]|$ is justified in three ways. Firstly, since classification algorithms typically work with real valued data, a rasterisation process would be required to take the real and imaginary parts of $X[k]$ into a single vector (or amplitude and phase). Since $X[k]$ is complex conjugate symmetric, there will still be redundancies that need to be removed. Second, taking the absolute value corresponds with removing the phase information of the frequency components. Loosely speaking, this means that using $|X[k]|$ as a representation for classification makes it invariant to shifts of the underlying time series. In other words, no effort is required to align the input ECG segments prior to processing, and the process removes information that is explicitly not wanted. Finally, due to symmetry of $|X[k]|$, only half of the samples are needed, resulting in the halving of the feature space dimension.

Therefore, *Spectra* refers to features obtained by computing the absolute value

of the discrete Fourier transform for a window of ECG, discarding the redundant samples. If $X[k]$ are the samples of the discrete Fourier transform of $x[n]$ of length N, then the Spectra representation is given as

$$\text{Spectra} = \left[|X[1]|, \ldots, |X[L]|\right], \; L = \left\lceil \frac{N}{2} \right\rceil \; . \tag{3.12}$$

As noted previously, many studies use a low feature dimension when performing classification. However the features themselves are often somewhat ad-hoc and choice of feature dimension is poorly justified. In order to realise some form of systematic dimension reduction, principal components (PCs) of the Spectra feature transformation is obtained. This is applied in a per category fashion to prevent the majority categories from dominating the learned basis functions. This must therefore be treated as a supervised step, as label information is used.

Assume a set, $\mathcal{S}$, of ECG segments for which the Spectra representation has been computed, a set, $\mathcal{L}$, containing corresponding category labels for each segment, and a set of the categories S contained in $\mathcal{L}$. Then for each category $s \in S$, the basis functions $\mathbf{B}^s$ are computed using PCA, but without mean subtraction,

$$\mathbf{B}^s = \text{PCA}\left(\mathcal{S}_i\right), \; \left\{i : \mathcal{L}_i = s\right\}, \forall s \; . \tag{3.13}$$

Since the columns of $\mathbf{B}^s$ are ordered in order of most variance explained to least variance explained for the data of each category, this needs to be taken into account when combining the basis functions from different categories to form a single set of basis functions. A way to approach this, is to interleave the first N

basis functions from each category;

$$\mathbf{B}_{im} = \mathbf{B}_{ij}^{S_k}, \quad \begin{cases} k \in \{1, \ldots, |S|\} \\ \{m : m = (j-1)|S| + k, \ j \leq N\} \\ \forall i, j \leq N \end{cases} \tag{3.14}$$

Then, the final set of basis functions $\mathbf{B}'$ is obtained using the QR decomposition

$$\mathbf{B} = \mathbf{B}'\mathbf{R} \ . \tag{3.15}$$

N should be selected such that $\left\lceil \frac{N}{|S|} \right\rceil \leq \text{Span}(\mathcal{S})$, because otherwise, such basis projections would be rank deficient (and the redundant dimensions removed by the QR procedure). With PCA, mean subtraction is usually performed because the mean of the data is not considered as a useful component. Since the basis functions are being derived using per-category data however, there may be some useful information provided by the mean direction of each category. Representations formed by this method are referred to as Spectra NPC, which in the case of three class classification between NVR, VT and VF, results in $3N$ dimensional space.

3.3 Evaluation procedures

One matter of importance is the exact procedure used for training and testing the methods to be evaluated. As noted in 2.2.4, many methods in the literature do not produce unbiased evaluations for various reasons, such as re-substituting training data for testing, or using preselected subsets of the data which are easily separable for both training and testing. The only data selection performed in this thesis is to exclude entire records which contain references to "ventricular

flutter". This is because ventricular flutter is an ill defined concept [11], which is not widely accepted. The alternative to exclusion was re-annotation, which is an activity to be undertaken by an expert community using consensus, and therefore is outside the scope of this thesis.

3.3.1 Main evaluation procedure

In this thesis, bootstrap resampling as described in 2.6.3 is used for evaluating distributions of accuracy. An 80%:20% split of the records is performed at each randomisation, to obtain training and testing sets respectively. The randomisation is performed with some constraints to ensure that a) the approximate proportion between categories does not vary too heavily in training and testing sets, and b) the category with the least amount of realisations is spread fairly uniformly throughout the whole randomisation. This is important because using cross validation for SVM hyper parameter selection would be problematic if some of the cross validation folds happen to omit any instances from a particular category, or the if skew between classes is significantly different.

The procedure for accepting or rejecting randomisations is described briefly. Given R records, assume column vectors $\underline{V}^s \in \mathbb{N}^R$ for each category $s \in S = \{NVR, VT, VF\}$ quantifying the number of samples per category in each record. For each $\underline{V}^s$ is a corresponding normalisation factor, $T^s = \sum_i \underline{V}^s_i$, and the cumulative vector $\underline{N}^s_i = \sum_1^i \underline{V}^s_i$. Then, for a given permutation $\mathbf{P}$, and Euclidean distance d the following conditions should be met:

$$\sum_{i=1}^{|S|} \sum_{\forall j>i} d\left(\frac{\mathbf{P}\underline{N}^{S_i}}{T^{S_i}}, \frac{\mathbf{P}\underline{N}^{S_j}}{T^{S_j}}\right)^2 < 2 \tag{3.16a}$$

$$d\left(\mathbf{P}\frac{\underline{N}^m}{T^m}, \underline{R}\right)^2 < 0.5, \; m = \arg\min_s T^s \; , \tag{3.16b}$$

where $\underline{R} = (1, \ldots, R)/R$.

These conditions ensure that the average proportion between classes over the distribution of records does not vary too much (3.16a), and that the category with fewest examples (VT) is distributed close to uniformly along the records (3.16b).

3.3.2 SVM training and hyper parameter selection

Of the methods considered in the preliminary studies, SVM requires particular attention for the training procedure. Recall that there are some hyper parameters for SVMs that require optimisation in order to obtain the best result. One approach might be to train the SVM on the entire training data and evaluate the chosen accuracy metric on the same training data, but this comes with a risk of overfitting and poor generalisation. Therefore, in order to mitigate this risk, and select hyper parameters that are more likely to generalise to unseen data, cross validation as in 2.6.2 is used for the hyper parameter selection procedure. The number of folds chosen is 5, since there is lower risk to overestimate generalisation ability with a smaller number of folds, and also the computational burden is reduced. The metric used for assessing cross validation performance is the Acc_{bal} metric from (2.24). The selected hyper-parameters are those which maximise this metric across all 5 test folds.

There are several ways to optimise the hyper parameters of SVMs. Recall from 2.5.3 that the multi-category problem with SVMs are to be approached using the error correcting output codes method. This requires training of several independent SVMs, in this case, 6. There are three possible approaches to hyper parameter selection;

i) every SVM in the error correcting code ensemble uses the same hyper parameters, and the chosen metric is evaluated for all test data

ii) each SVM can optimise the hyper parameters independently, using only testing data with the labels that the classifier is intended to handle, or

iii) the hyper parameters for each SVM are optimised jointly, so that each classifier may have different hyper parameters that are selected to perform optimally given the entire error correcting code procedure.

Although optimal, the final option is computationally expensive, since it effectively raises the number of grid points to the power of the number of classifiers. Even with very small grids, or a small number of classifiers, the computational cost is clearly infeasible. The second option is no more computationally expensive than the first, and may improve the final result; therefore this approach to optimising the hyper parameters is used.

Since the record order is already randomised for the bootstrap resampling procedure, the records are not required to be randomised again. Thus, the cross validation folds are formed by records, by simply taking roughly equal amount of records in the order they are present in the training set. Then each classifier is trained for each point on the hyper parameter grid using all but the one fold held out for testing. This is repeated for each fold, and the position on the hyper parameter grid with the highest average accuracy metric across all folds is found. Then, the classifiers are trained using the entire training data set with the selected hyper parameters.

For linear, polynomial, and RBF kernels, there are some hyper parameters for which an appropriate range needs to be selected in order to form the grid to search along.

One parameter common to all methods is $\mathcal{C}$. This parameter is typically searched over a logarithmic range [34, 59]. It can be shown that an upper bound on generalisation error exists if $\mathcal{C} \geq 1/p$ [59], where p is the number of training examples. Therefore a suitable logarithmic range which starts at maximum

regularisation and decreases the amount of regularisation would be

$$\mathcal{C} = \frac{10^N}{p},\ N \in \{0, \ldots, 4\}\ . \tag{3.17}$$

For selection of γ for the RBF kernel, a couple of well-known heuristics exist;

1. Select γ as $1/l$ [60], where l is the dimension of the space.

2. Compute pairwise distances d_{ij} for some subset of the training data (or entire dataset if it is small enough), and compute γ based on 5% and 95% quantile statistics [61] in order to have a kernel that is neither too inclusive of all points, nor too exclusive.

A variation on the quantile method is used to form the search range for γ that is simpler to compute. For each training sample, its squared Euclidean norm is computed, and then D_{mean} is the average of all these values. Then, a logarithmic search range for γ is given as

$$\gamma_{search} = 10^N, N \in \{\gamma_{start} - 2, \ldots, \gamma_{start} + 2\}\ , \tag{3.18}$$

where

$$\gamma_{start} = -\log_{10} D_{mean}\ . \tag{3.19}$$

For the polynomial kernel, there are three parameters, two of which require tuning. Recall that the polynomial kernel is given as

$$K_p(\underline{x}, \underline{y}) = (k + c\langle \underline{x}, \underline{y} \rangle)^d\ ,$$

where k, c, d are all selectable. Common choices for k are 0 and 1. Here, $k = 1$ is selected to allow for the full diversity of cross terms. Then only the selection of c and d remains. Intuitively, c can be set according to the type of response desired

from the kernel function. Given $k = 1$, with an initial estimate for c given as

$$c_{start} = \frac{1}{\max ||\mathcal{D}_i||^2} \ , \ \forall i \tag{3.20}$$

where $\mathcal{D}_i$ are training data points, then, the kernel takes values

$$K_p\left(\mathcal{D}_i, \mathcal{D}_j\right) \in \left[0, 2^d\right] \ , \tag{3.21}$$

for training data. For testing data this range may be exceeded, but if the training data is representative, it should not be significant. However this may be too restrictive. Therefore, c may be searched for over a range allowing the inner product to become larger, i.e.

$$c = c_{start} \cdot 2^N, \ N \in \{0, 0.5, \ldots, 2\} \ . \tag{3.22}$$

Then the kernel values for training data pairs is in the range

$$K_p\left(\mathcal{D}_i, \mathcal{D}_i\right) \in \left[(k-c)^d, (k+c)^d\right] \ . \tag{3.23}$$

From this construction it is easy to see two things. First, pairs of points with positive inner products will have more impact that pairs with negative inner products. Second of all, uncorrelated and short points will take similar kernel values, which is useful since, for example, a diastolic pause in the ECG may look like VF, but it is not. In order to increase differentiation between point pairs with negative inner product from those with positive inner products, odd values for d should be used. To prevent the size grid to be searched from becoming too large, $d = 5$ is fixed a priori.

3.4 Assessments and analysis

3.4.1 Experiments

Classification experiments were performed using 50 bootstrap resamples for test estimates, and five fold cross validation for hyper-parameter selection. In order to have fair comparison between methods, each experiment used the same resamples. Table 3.1 shows the different combinations of experimental parameters (representation space, observation length and classifier), which amounted to 120 distinct experiments. ECG segments obtained from each record for training and testing were non-overlapping. The purpose was to investigate the impact of segment size and classifier, and to understand the impact of dimension reduction on classification performance. Test scores are given as balanced accuracies, i.e. Acc_{bal} from (2.24), and in some cases, sensitivities of each category and the confusion matrix averaged over all bootstrap resamples.

Apart from confusion matrices, full distributions of each metric are presented, in the form of boxplots, which allows visualisation of the inter-quartile range and median of a given metric. Additionally, notches around the median are visualised which allows to conduct a visual statistical test of whether two medians are significantly different at the 95% confidence level [62], a useful indicator for determining whether two methods perform significantly differently.

The combination of 2 s ECG segments classified with an RBF SVM and Heur2 representation corresponds with the experiments conducted in [31], while the combination of 8 s ECG segments classified with an RBF SVM and Heur8 representation closely resembles the experimentation performed in [32]. Particular attention is paid to these combinations, as they serve as the baseline for comparisons.

In neither case was the correspondence between the experiments conducted in

Table 3.1: The investigated parameters varied in experimentation for this chapter are observation length, classifier types and representation spaces. For each of these experimental parameters, this table lists all the possible values. The result is a total of 120 different experiments

Parameter	Values
Observation length	1 Second, 2 Seconds, 4 Seconds, 8 Seconds
Classifiers	RBF SVM, Polynomial SVM, Linear SVM, LDA, QDA
Representations	Spectra, Spectra 15PCs, Spectra 10PCs, Spectra 5PCs, Heur8, Heur2

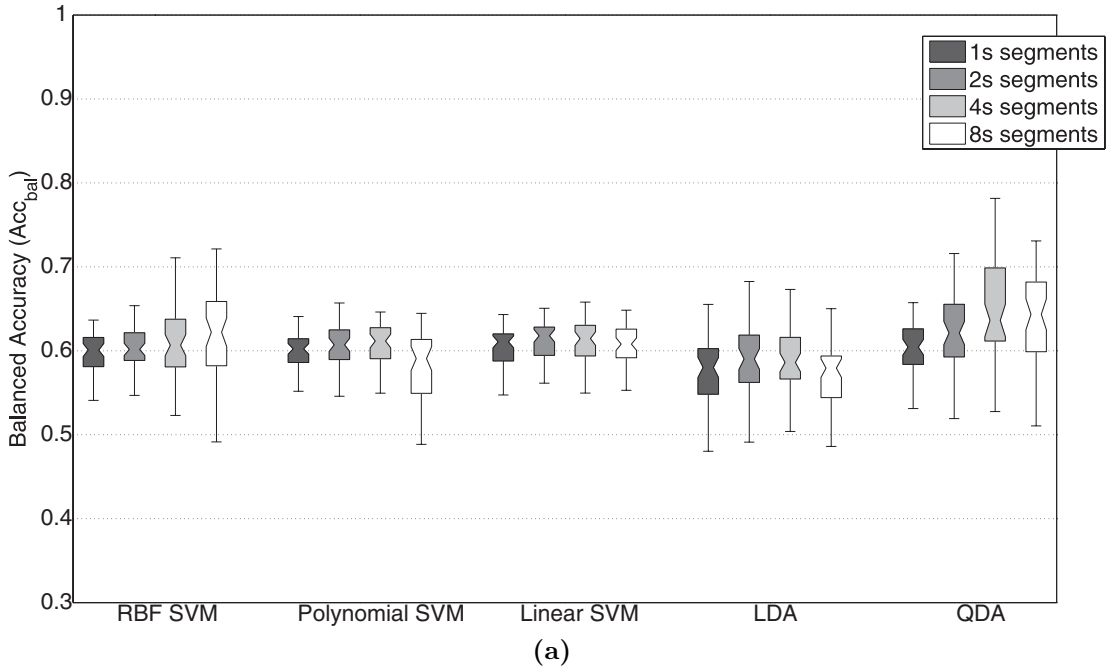
this thesis and the original studies exact. This is due to different databases, different considered classification task (3-way classification), and different procedure for hyper-parameter selection. However, all these choices amount to performing a more thorough and realistic assessment of these methods.

3.4.2 Results

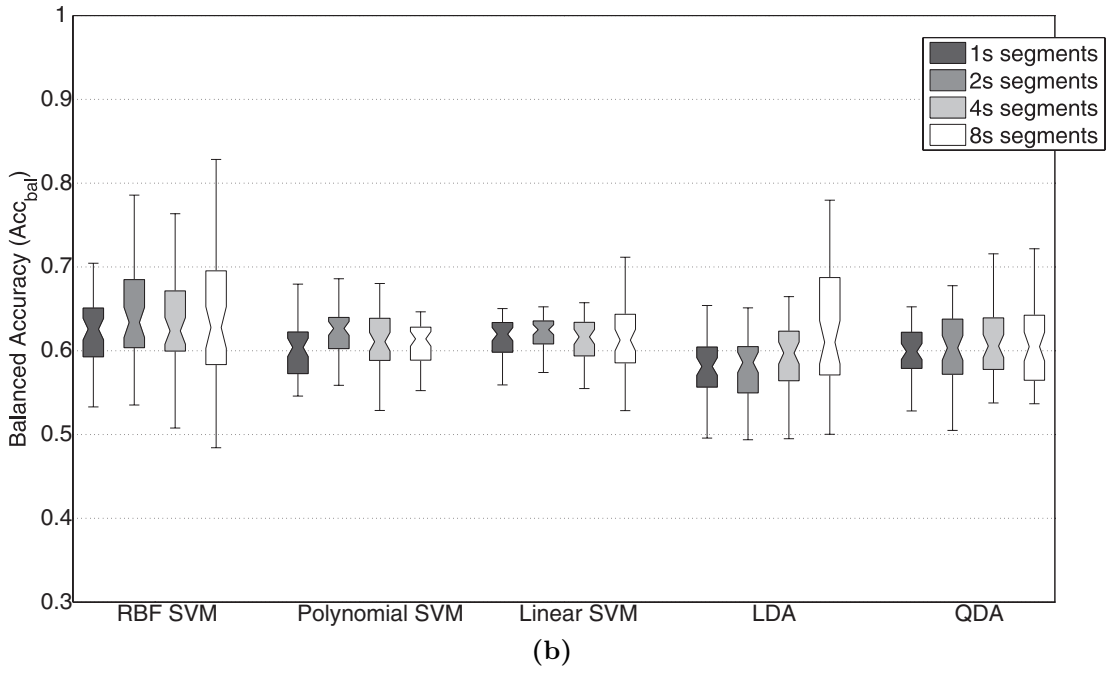
Figure 3.1 shows the distributions of the Acc_{bal} metric across 50 bootstrap resamples using the benchmark features Heur2 and Heur8, for all the investigated classifiers and ECG segment lengths (3.1a and 3.1b respectively).

It can be seen from Figure 3.1a that for the Heur2 representation space, the choice of classifier and observation length did not have much impact on the overall classification accuracy. The polynomial SVM classifiers did not do any better than linear SVM, however, QDA was the best performing method when combined with 4 s segments. Overall, most parameter combinations did not perform much better than median accuracy around 60% for the Heur2 representation space.

On the other hand, the median accuracy for the Heur8 representation, as shown in Figure 3.1b, for many different classifiers was slightly higher. In particular, for almost all segment lengths, Heur8 with an RBF SVM classifier performed with a typical median accuracy of 62%–63%. Again, there was not much difference in performance between polynomial and linear SVMs, while the performance for LDA was less than that of linear SVM, with the exception of 8 s



(a)



(b)

Figure 3.1: Acc_{bal} distributions for all classifiers and segment lengths with (a) Heur2 representation space and (b) Heur8 representation

segments, while QDA performed at a level similar to polynomial SVM.

Figure 3.2 shows the distributions of the Acc_{bal} metric for the Spectra based classifiers. The bootstrap resamples were the same as those for testing Heur2 and Heur8. Each subfigure represents a single observation length, 1 s, 2 s, 4 s and 8 s (3.2a, 3.2b, 3.2c, 3.2d, respectively). The spectra representations are presented in order of decreasing dimension, from Spectra, to Spectra 5PC, and grouped by classifier.

A principal observation to be made from these data, is that dimension reduction performed systematically had little impact on classification accuracy. In most cases, reducing the dimension of the Spectra representation via PCA did not result in a noticeable decrease in accuracy. Dimension reduction improved the performance of the QDA classifier in most cases, however, apart from the case of 2 s observation segments, the improvement was not any better than other classifiers, and in general QDA performed much worse. As the segment length increased, the polynomial kernel also performed poorly. The LDA and linear SVM classifiers performed similarly, with linear SVM giving a slight improvement over LDA. Classification using RBF SVMs in Spectra and all dimension reduced spaces was consistent (67% median) for all segment lengths.

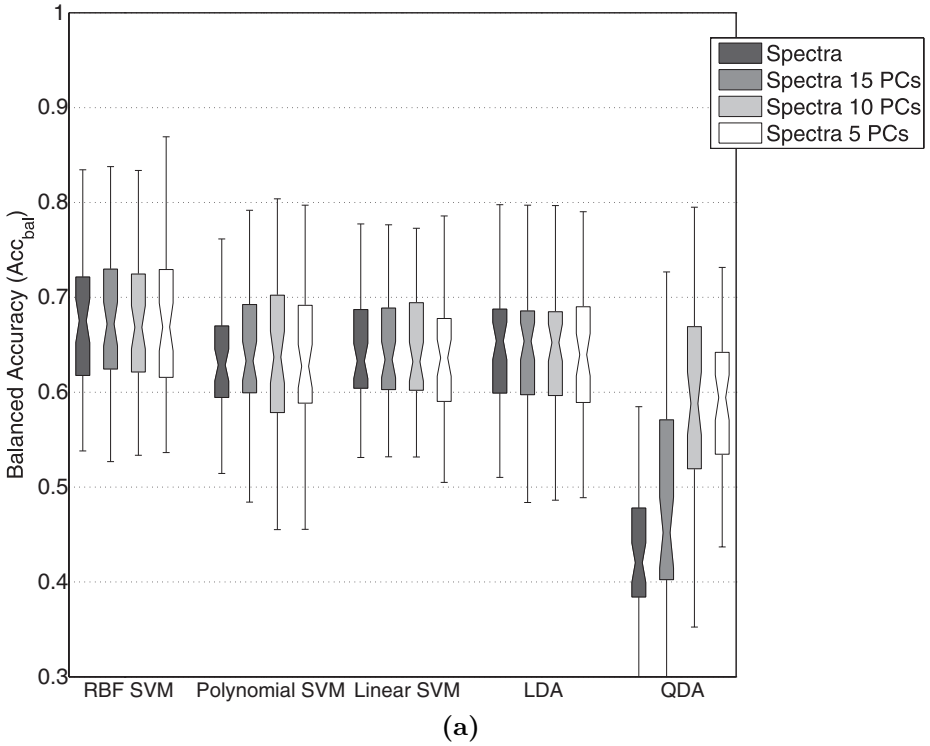
Figure 3.3 shows sensitivities of each of NVR, VT and VF for Heur2 (Figure 3.3a) and Heur8 (Figure 3.3b) representations and classifier combinations, with the best median Acc_{bal} scores. These include the reference methods Heur2 2 s classified with RBF SVM, and Heur8 8 s classified with RBF SVM.

Finally,the sensitivities of each category are shown in Figure 3.4 for some selected methods, these are;

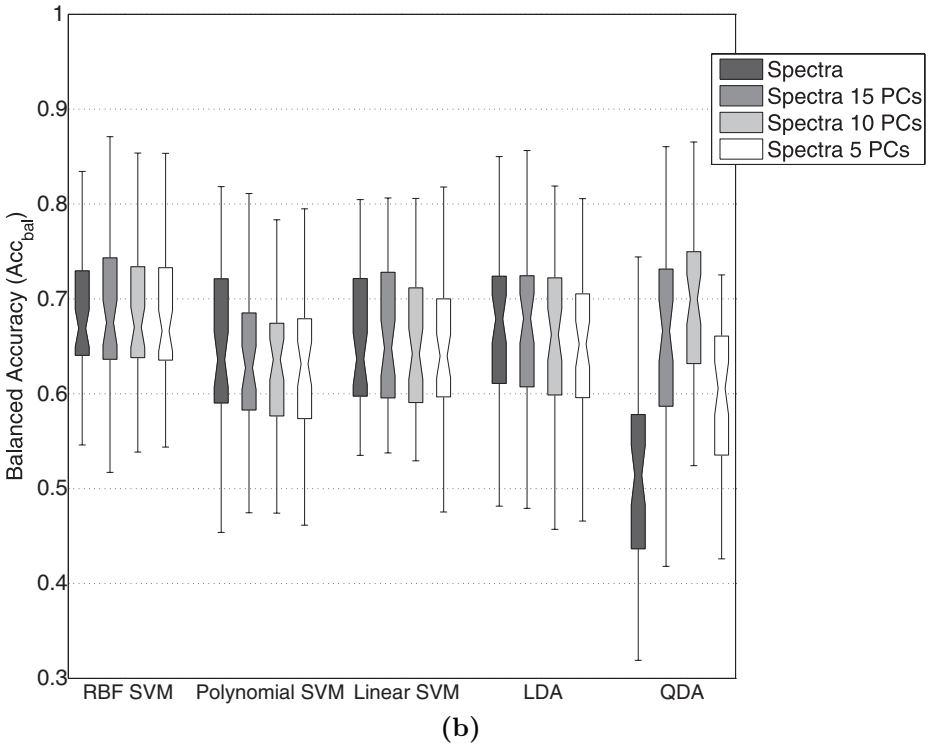
1. The Heur2 2 s RBF SVM reference combination

2. The Heur8 8 s RBF SVM reference combination

3. Spectra 1 s–8 s RBF SVM (Spectra NPC is omitted since Acc_{bal} distribu-

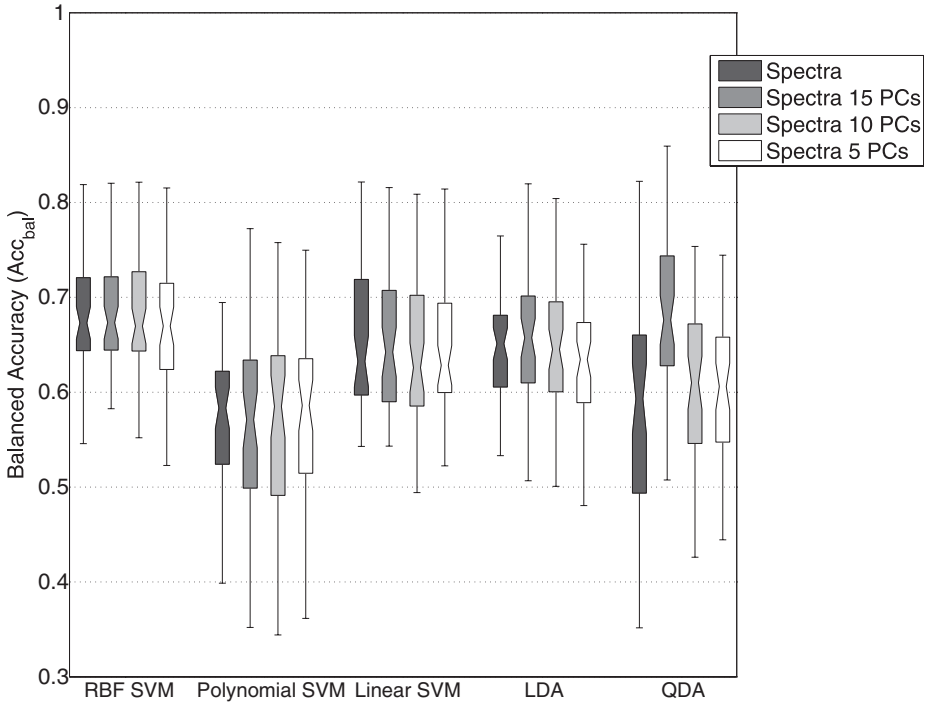


(a)

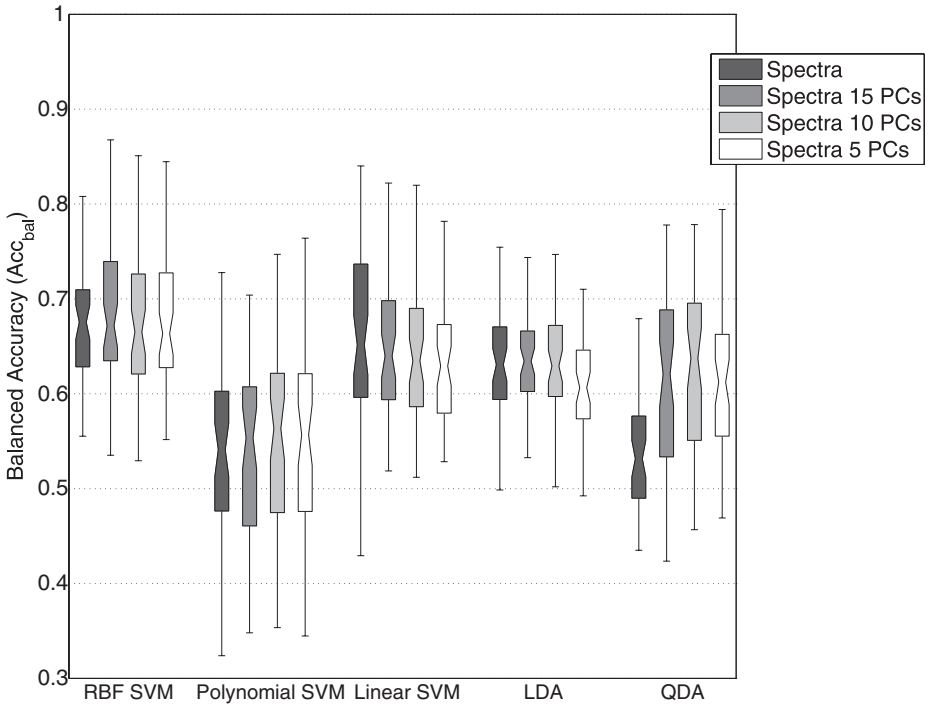


(b)

Figure 3.2: Acc_{bal} distributions for all classifiers and Spectra / Spectra NPC representation spaces for (a) 1 s segments, (b) 2 s segments

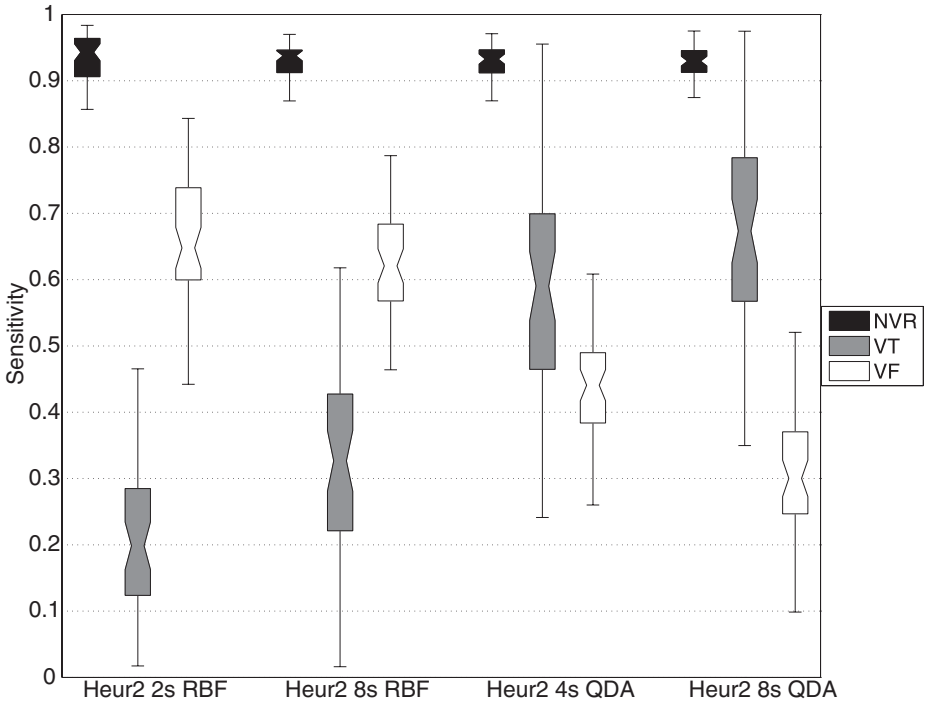


(c)

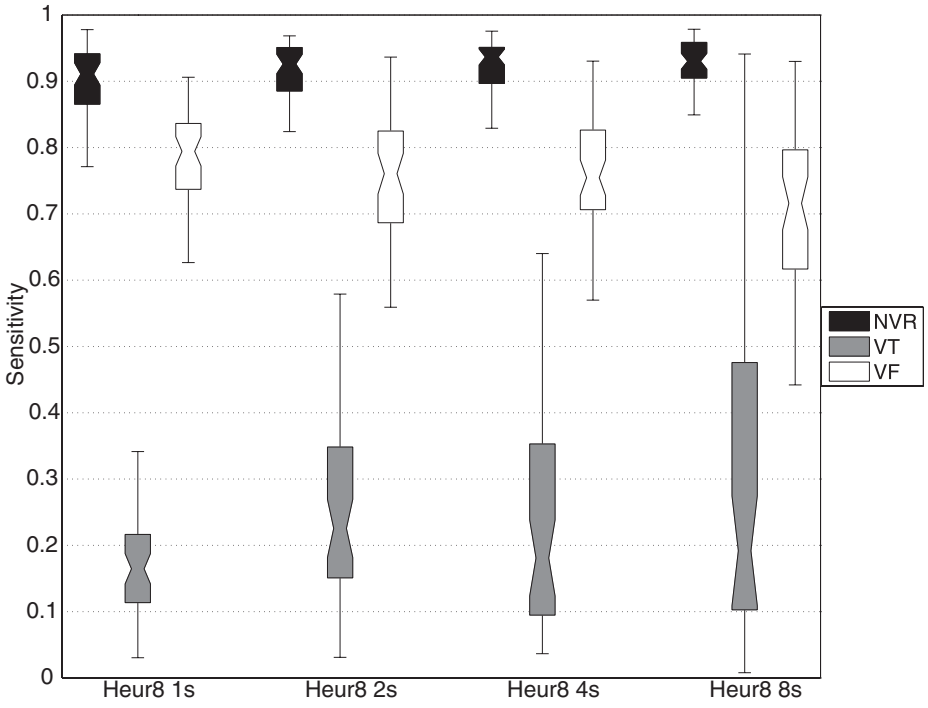


(d)

Figure 3.2: Acc_{bal} distributions for all classifiers and Spectra / Spectra NPC representation spaces for (c) 4 s segments, (d) 8 s segments



(a)



(b)

Figure 3.3: Sensitivity distributions of each category for (a) Heur2 representation space with selected segment lengths and classifiers, and (b) Heur8 representation space classified with an RBF SVM and all investigated segment lengths

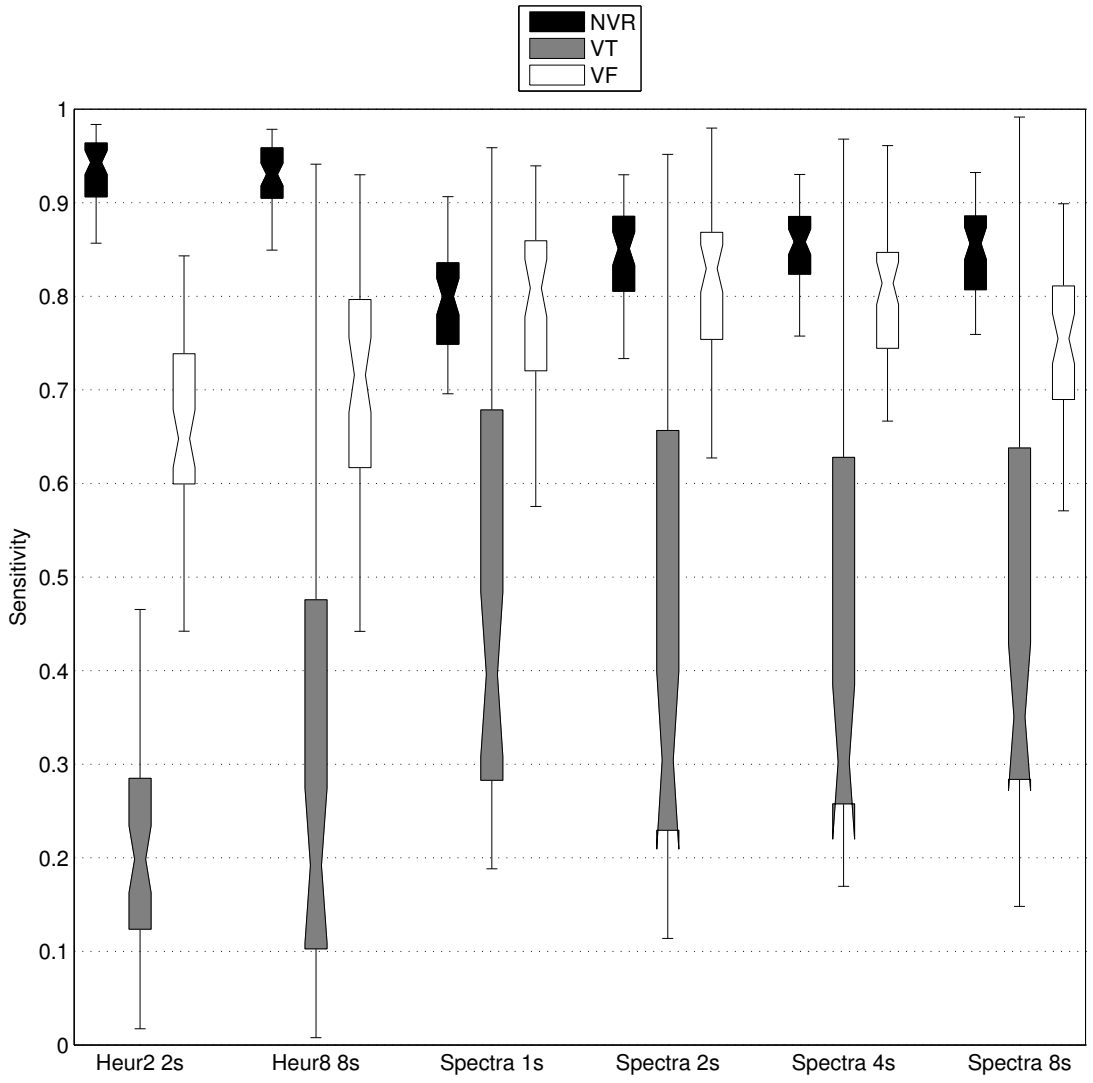


Figure 3.4: Distributions of sensitivities for each of NVR, VT and VF for the Heur2 and Heur8 reference methods, and Spectra representation spaces for all investigated segment lengths. All were classified using the RBF kernel SVM

tions were not significantly different).

In all cases, these results are shown for RBF SVMs because this classifier performed consistently well across representations and observation lengths. In order to obtain further insights, the average confusion matrices over all the bootstrap resamples are also shown in Table 3.2 for each of these methods.

Table 3.2: Average confusion matrices over all bootstrap resamples for each classifier method shown in Figure 3.4. Rows are the ground truths, and columns are the diagnoses made.

		Diagnosed as		
Method	**Ground Truth**	**NVR%**	**VT%**	**VF%**
Heur2 2s	NVR	93.5	4.0	2.5
	VT	19.8	20.4	59.8
	VF	10.2	23.4	66.4
Heur8 8s	NVR	92.1	5.5	2.4
	VT	13.4	31.0	55.6
	VF	8.0	21.2	70.7
Spectra 1s	NVR	78.5	6.4	15.1
	VT	16.5	47.5	36.0
	VF	4.7	16.5	78.8
Spectra 2s	NVR	83.6	3.6	12.8
	VT	19.9	41.7	38.4
	VF	4.7	14.7	80.7
Spectra 4s	NVR	84.6	3.3	12.1
	VT	18.8	42.7	38.4
	VF	5.1	14.9	80.0
Spectra 8s	NVR	84.2	4.0	11.8
	VT	17.4	45.7	36.9
	VF	5.2	19.8	74.9

3.4.3 Discussion

For the 120 experiments performed, all data were shown using the Acc_{bal} summary statistic. Although not completely informative, it was useful enough to identify trends among classifiers, segment length and representation spaces.

The most prominent observation was that the RBF SVMs performed consistently as the best or near best classifiers. Linear classifiers performed reasonably consistently, although the generative LDA performed slightly worse than the discriminative SVM. QDA classifiers produced some of the highest results, but this effect was not consistent with varying observation length and representation space pairs. This was probably due to the generative nature of QDA and the fact that as segment length increased, less training data was available. Classification results with the polynomial kernel were generally poor, and this was probably due to the difficulty in optimising so many different kernel parameters appropriately.

Note that the dimensions of the Heur2 and Heur8 representation spaces were 2 and 8, respectively. In comparison, the minimum proposed dimension in this investigation was 15, and for almost all combinations of segment length and classifier better Acc_{bal} distributions were obtained. This suggests that the dimension reduction employed by previous studies is far too aggressive. However, distributions of individual category sensitivities showed a slightly different picture. Figure 3.4 showed that the Heur2 and Heur8 representation spaces perform consistently well at detecting NVR, which makes sense as these features appear to have been engineered for detection of arrhythmia presence, rather than differentiation between arrhythmias. Interestingly, the lower dimensional Heur2 obtained higher median NVR sensitivity than Heur8 (when configured with experimental parameters similar to those of the original study), however the Heur8 representation had a slightly better ability to differentiate between VT and VF. Additionally, according to the confusion matrices shown in Table 3.2, Heur8 representation

made less incorrect assignments of ventricular arrhythmias to the NVR category. Spectra representations by comparison, made less incorrect assignments of VF to NVR, but the average detection rate for NVR was much lower than for Heur2 or Heur8 classifiers. In general, the performance of Spectra based classifiers did not change significantly with variations in observation length for the RBF SVM classifiers. Overall, with the exception of a few QDA classifiers which were not generally consistent, the best Acc_{bal} distributions were obtained when classifying Spectra 1 s observations with an RBF SVM. In particular, this combination obtained the best VT and VF sensitivities.

An important point to note, is that in Figure 3.4, some of the boxplot notches extend beyond the lower quartile for the VT sensitivity distribution. This is because the sensitivity of this category varies highly across the different bootstrap resamples, indicating that 50 resamples is not sufficient for good estimates of the distributions. Therefore, in later chapters, a higher number of bootstrap resamples will be considered.

3.5 Summary and conclusions

An initial assessment on suitable experimental parameters was performed, using a set of 120 experiments generated by varying observation length, representation space, and classification algorithm. Two sets of feature spaces from previous studies [31, 32] were included in the choice of representation spaces investigated in order to determine their discriminative capability and assess the impact of dimension reduction. Other representation spaces were formed from Fourier magnitude spectra and their reduced dimensional spaces. It was found that systematically reducing the feature dimension via PCA does not have an impact, down to 15 dimensions, which is still considerably higher than the comparative works.

The features developed in prior studies, some of which are designed for discriminating between NVR and ventricular arrhythmias, and others for differentiation between NVR, VT and VF were shown to be less capable on average at discriminating between VT and VF, with a prominent bias towards classifying all ventricular arrhythmias as VF. On the other hand, these feature spaces were demonstrated to be very capable at detecting NVR, although the higher dimensional spectral features proposed in this chapter diagnosed less ventricular arrhythmias as NVR.

The impact of ECG segment length for analysis was also assessed, although not as thoroughly as in [31], and it appeared that increasing the segment length has minimal impact on the ability to discriminate between VT and VF.

It was evident from the distributions of per category sensitivities that the biggest challenge was correctly identifying VT. While notable improvements were obtained by using the higher dimensional Spectra representation, this came at the cost of reduced capability to correctly identify NVR. Additionally, patient variability appears to contribute significantly to classification accuracy, since the typical interquartile range on overall accuracy was between 7%–10%. Therefore, another objective is to reduce the variability of diagnostic accuracy.

Chapter 4

Ensemble methods and temporal ensembles

In the previous chapter, investigations were performed using a variety of observation window lengths, classification algorithms and representation spaces. However the observation windows considered were not overlapping, so a decision was output only once per observation segment. In this chapter analysis is performed using overlapping observation windows using the best classifiers identified in the previous chapter, and ensemble methods to improve accuracy. Instead of simply classifying each segment, a group of segment outputs are used together to produce a decision. It will be shown that this can improve classification accuracy for all representation spaces. Additionally, the strengths of different representation spaces to identify particular categories are utilised by combining trained classifiers in order to obtain jointly better sensitivities for all three categories NVR, VT and VF than with any of the individual classifiers considered.

4.1 Chapter outline

Previously, three way classification between NVR, VT and VF was investigated. This was performed using bootstrap resampling for error rate estimation. A variety of classification algorithms were tested, and representations included Spectra, Heur2 and Heur8 formed from varying observation lengths. Generally speaking, VT sensitivities were low, and VF sensitivities were moderate. In this chapter, methods for forming decisions from multiple classifiers and multiple classification outputs are examined, with the aim of improving VT and VF sensitivities.

First, a brief overview of ensemble methods is given. Error correcting output codes is considered as an ensemble method, and the principle of stacked generalisation is elaborated. Then, given the SVM classification framework with error correcting codes, a scheme is proposed for what is effectively temporal averaging of decision values in order to improve classification accuracy. A method is proposed for hierarchical classifier combining in order to capitalise on the ability of particular representation spaces to correctly identify particular categories. In order to remove the reliance on arbitrary functions for error correcting codes and temporal averaging techniques, stacked generalisation is proposed for combining these two concepts optimally. Finally, these concepts are tested and experimental results are provided.

4.2 Overview of ensembles

The general idea behind ensemble methods is to have more than one function generating outputs given the input, and combine the outputs in such a way that on average, the output resulting from the combination is better than the average output from any of the individual functions. This allows for combination of

arbitrarily many weak classifiers into a single strong classifier.

There are many different approaches to forming ensembles of classifiers. An overview of a large variety of methods is presented in a review [63] of ensembles for classification. A couple of the most prominent techniques are mentioned briefly.

1. Adaptive boosting [64] works by weighting each training point according to its difficulty to classify. Repeatedly misclassified points accumulate a higher weight than correctly classified points, causing the successive classifiers to be generated to focus on classifying these points. Then, each classifier is additionally assigned a weight. Outputs of the learned classifiers are combined according to their weights in order to make predictions. This is a popular form of a more generic construct generally known as boosting, but these type of techniques fail in situations with even a small amount of label noise [65].

2. Bootstrap aggregating, or bagging [66] works by sampling from the training data set, with replacement, and building a classifier for a given sample. A number of samples are taken and classification is performed by voting among all the learned classifiers. It is noted that this procedure is designed to work best with classification algorithms which are unstable, i.e. a small change in training data causes a large change in the estimated function. The idea is that combining outputs of many unstable classifiers, each with a high variance, reduces the variance of the overall ensemble of classifiers.

3. Random subspace ensembles, or attribute bagging [67, 68] are ensembles formed by sampling from the set of predictors, and building a classifier using only the selected predictor subset. This process is repeated with many random predictor subsets. Outputs from each classifier are combined by voting or weighted voting. This technique is related to the random forests technique [69], which uses modified variant of classification trees to select

subsets of predictors at the branch splits, and uses bootstrap aggregation to form an ensemble of many such trees.

4. Stacked generalisation [70] forms an ensemble of classifiers from a cross validation procedure. Another classifier (or generaliser) is trained over the decisions of each member of the ensemble for the training data points, in order to learn and correct for the individual classifier biases. Alternatively, a classifier may be trained over the output decision values [71], whether they be conditional probabilities, or some other scores such as hyperplane distances in SVM models.

4.2.1 SVMs with error correcting codes

It turns out that the error correcting output codes approach to multiclass classification with SVMs is actually an ensemble method. Recall that an SVM deals with a binary classification problem. Therefore, categories are grouped together for the training procedure in varying combinations. Each one of these groupings forms a single member of the ensemble committee. This form is represented by a coding matrix $\mathbf{W}$ with each element 1 or -1. A specification for the class grouping of each classifier in the ensemble, f^n, is given in the columns of $\mathbf{W}$. In the original formulation [44], the assignment should be made in order to maximise the Hamming distance between rows of $\mathbf{W}$, and each category is assigned to a row. For prediction, each member of the ensemble is tested for its response, and a column vector, $\underline{V}_n = \mathrm{sgn}\left(f^n\left(\underline{x}\right)\right)$, of these decisions is formed. The assigned class is then given by

$$C\left(\underline{x}\right) = \arg\min_m \mathcal{H}\left(\mathbf{W}_{mn}, \underline{V}\right), \tag{4.1}$$

where $\mathcal{H}$ is the Hamming distance function.

A modification was proposed [45], to allow sparse codes (0's in the coding matrix) and use the output values rather than decisions, by replacing the Hamming distance with a generic loss function χ,

$$C\left(\underline{x}\right) = \arg\min_{m} \sum_{n=1}^{N} \chi\left(\mathbf{W}_{mn} f^{n}\left(\underline{x}\right)\right) \ . \tag{4.2}$$

Therefore, with error correcting output codes, the decision criterion is minimisation of a loss function. A drawback of this form is that the loss function is user selected and thus there is no proper adaptation of the loss function to the problem.

4.2.2 Stacked generalisation

In general, most classification algorithms have some intermediate values or scores that are used to obtain the final decision. These are often class conditional probabilities, but may also be hyperplane distances or some other scores. In the case of SVMs, these are hyperplane distances. During N-fold cross validation, N classification models are built using subsets of the data, and these are tested using the held out fold of data. A common strategy when using cross validation for model building is to select the model which performs best when evaluated upon its corresponding test fold, but this can be problematic for a couple of reasons. The test fold may happen to be conveniently good and such a classifier may not learn from an important subset of the data. Stacked generalisation [70, 71] proposes to increase classification accuracy by retaining all of the learned models and evaluating the entire data set using each model. Then, for each training data point there are decisions or output values from each of the learned models, one of which is from a model that was not exposed to the given data point for its training procedure. All of the outputs or decisions can then be combined to form a new vector of training points, and these can be used to train a new classifier,

which can adaptively weight each classifier output in order to correct for biases and obtain better results than would be possible with just the best classifier.

Stacked generalisation need not be performed with several models that are obtained using cross validation. They may be trained using bootstrap resampling, data hold out, different base learners or models with different hyper parameters. In the case of SVMs with error correcting output codes, these may be trained using a portion of the training data and the higher level classifier is trained using the outputs generated by the ensemble of SVMs on a superset of data, i.e. data with some unseen examples. For SVMs trained using data subsampling for class balancing, the entire training data may be used for learning the higher level generaliser particularly as many training samples will have been discarded by the subsampling process.

4.3 Proposed ensemble methods

As mentioned previously, ensembles can be constructed in many ways [63]. In this section some ensembles are described which leverage knowledge of the problem in order to improve classification results.

4.3.1 Ensembles of SVM decisions over time

One way to form an ensemble that capitalises upon the structure of the problem at hand is to form ensembles temporally. In Chapter 3, shorter windows of ECG were observed to be equally capable in discriminating between all three categories as longer windows of ECG. Recall the decision rule (4.2) when using the error correcting codes method for multiclass SVMs. Assuming an M class problem, rather than using the decision value of each SVM for a single window of ECG, consider vectors of decision values from T, possibly overlapping, temporally local

segments of ECG, $\underline{x}^t$, i.e.

$$\underline{f}_t^n = f^n(\underline{x}^t),\ \forall t \in [1, \ldots, T]\ . \tag{4.3}$$

Then, using an aggregation function $A : \mathbb{R}^T \to \mathbb{R}$ and combining with (4.2) the decision rule becomes

$$C(\underline{x}) = \arg\min_m \sum_{n=1}^{N} \chi(\mathbf{W}_{mn} A(\underline{f}^n))\ . \tag{4.4}$$

Some choices for A include

$$A\left(\underline{f}\right) = \begin{cases} \text{mean}(\underline{f}), & \\ \text{median}(\underline{f}), & \\ \text{mode}(\text{sgn}(\underline{f})), & \text{majority vote} \\ \underline{f}_{\ \arg\max_t |\underline{f}_t|}, & \text{maximum absolute value} \end{cases} \tag{4.5}$$

Alternatively, temporal ensembles may be formed from the loss values after decoding rather than the SVM outputs. Consider a similar aggregation function $B : \mathbb{R}^T \to \mathbb{R}$, and formation of a matrix of loss values,

$$\mathbf{L}_{tm} = \sum_{n=1}^{N} \chi\left(\mathbf{W}_{mn} f^n\left(\underline{x}^t\right)\right),\ \forall t, m\ . \tag{4.6}$$

Then, the decision rule can be expressed as

$$C(\underline{x}) = \arg\min_m B\left(\mathbf{L}_{tm}\right),\ \forall t\ , \tag{4.7}$$

except in the case of majority voting, where the aggregation function B is the

mode function, and is applied slightly differently, i.e.

$$C(\underline{x}) = \text{mode}\left(\arg\min_{m} \mathbf{L}_{tm},\ \forall t\right) \ . \tag{4.8}$$

Some choices for B include

$$B\left(\underline{f}\right) = \begin{cases} \text{mean}(\underline{f}) \\ \text{median}(\underline{f}) \\ \min(\underline{f}) \end{cases} \tag{4.9}$$

This family of schemes for making decisions which incorporates temporal SVMs outputs with error correcting output codes is named as local context ensembles (LCEs). An important point to note about LCE decoding is that it can be performed either forwards or backwards temporally. In case that it is performed forwards, a latency is incurred equal to the total duration used for aggregating. Thus, in the context of a real-time analyser, a decision for a specific point in time when aggregating over 8 s is not made until 8 s of ECG is acquired, even if the base observation length is only 1 s. On the other hand, when using backwards decoding, the amount of data that needs to be acquired is only the base observation length, which has the potential to significantly reduce the amount of time required to make a decision.

4.3.2 A hierarchical approach to decision making

In the scenario that some classifiers are better at certain subtasks than others, and the label structure has a hierarchical taxonomy, it may be worthwhile considering decision making via a hierarchical structure. This type of approach is considered by some for multiclass SVM classification schemes [42,43]. Results from Chapter 3 demonstrated that no single representation investigated was effective at detecting

all categories. Of note was the fact that the lower-dimensional Heur8 and Heur2 feature sets obtained good NVR sensitivities. On the other hand, Spectra representation obtained better VT and VF sensitivities.

Therefore it makes sense to try to capitalise upon this in order to improve overall classification accuracy. For this purpose, consider that for a given segment of ECG, two decisions (each of NVR, VT or VF) are made simultaneously by classifiers, C_1 and C_2, induced on different sets of features. Let C_1 be the primary classifier. If a NVR decision is taken by the primary classifier, then that decision is accepted, otherwise the decision of secondary classifier C_2 is accepted:

$$C(\underline{x}) = \begin{cases} C_1(\underline{x}), & C_1(\underline{x}) = \text{NVR} \\ C_2(\underline{x}), & C_1(\underline{x}) \neq \text{NVR} \end{cases} \tag{4.10}$$

C_1 and C_2 can be composed in many ways, by varying both the learner and transformation functions. For example, C_2 can be restricted to making only VT or VF, or it may be allowed to make a NVR decision. The hierarchy may also be constructed with the master decision making process based on a category other than NVR. In this case however, there is a very obvious hierarchical structure, in that VT and VF are both ventricular arrhythmias. That they share so much in common is what makes the decision task difficult. Therefore this structure is selected and fixed a priori.

4.3.3 Stacking with SVMs

One of the potential issues with the LCE method described in 4.3.1 is that the aggregation function is simply chosen among the best of a limited prior selected set. In reality, there are uncountably many aggregation functions, and loss functions for error correcting output codes. However since classification is essentially a function estimation technique, and the learned SVMs will have

biases, it makes sense to use the stacked generalisation ensemble technique to jointly correct for the biases in the learned SVMs whilst simultaneously learning an appropriate combiner (rather than error correcting code loss function) and temporal aggregation function. Importantly, and usefully, the error correcting code ensemble member weights need not be learned disjointly from the temporal component weights. Thus, instead of testing an ad-hoc selection of aggregation and loss functions, prior beliefs about the best form of combiner can be embedded in the selection of the classification algorithm used as the higher level classifier.

Recall that the basic principle of stacking is to generate outputs from a trained model, or set of trained models, including some data that the models were not exposed to for their training phase. Then, these outputs are used to train another classifier which adapts to the biases of each base classifier function in order to learn an overall stronger classifier.

Consider a set of transformed data, $\mathcal{D}$, with dimension P, which is to be used for training a classification model. Then, for an N-fold cross validation procedure where $\mathcal{D}$ is split into N non-overlapping sets $\mathcal{D}^n$, the training data set for the n^{th}-fold is denoted as $\hat{\mathcal{D}}^n = \mathcal{D} \setminus \mathcal{D}^n$. With a learning function L, a model is learned with each training data subset,

$$f^n = L\left(\hat{\mathcal{D}}^n\right), \ \forall n \in \{1, \ldots, N\} \ , \tag{4.11}$$

where $f^n : \mathbb{R}^P \to \mathbb{R}^K$, i.e. f^n outputs a row vector with dimension K given an input data point of dimension P. Then, a new training data set $\mathcal{D}'$ is obtained by concatenating the outputs of each model evaluated at each data point, i.e.

$$\mathcal{D}'_i = [f^1(\mathcal{D}_i), \ldots, f^N(\mathcal{D}_i)], \ \forall i \ . \tag{4.12}$$

Different parts of the dataset will be unseen by different models, and it is this

which allows the top level classifier to learn about the bias of the individual classifiers. Then, in order to capture local temporal variation on the classifier outputs, each data point is further augmented by including the last T outputs

$$\overline{\mathcal{D}}_i = [\mathcal{D}'_i, \ldots, \mathcal{D}'_{i-T+1}], \ \forall i \ . \tag{4.13}$$

The procedure to obtain $\mathcal{D}'$ and $\overline{\mathcal{D}}$ for testing data is identical to that for the training data.

An important consideration is the resulting dimension of the augmented data set. In the case of an error correcting code 3 class SVM with 6 binary classifiers, 5-fold cross validation for model diversity and using the last 15 temporal outputs, the resulting feature dimension is $6 \cdot 5 \cdot 15 = 450$. This is problematic as there may not be enough data to learn from in such high dimensions. There are a few possibilities for keeping the dimension under control.

A) Perform error correcting code decoding and build ensembles of loss values. For the three class case, this reduces the dimension by a factor of 2. This comes with the disadvantage that an arbitrary combiner must still be used at the loss decoding stage.

B) Use fewer folds for cross-validation. The minimum reasonable number is 3 folds, since the amount of training data available to the lower level SVM classifiers is reduced (5 fold uses 80% of the data, while 3 fold uses 66% of the data). The reduction in dimension is 40% compared to 5-fold cross validation.

C) Do not use cross-validation at all. Instead, (ab)using the fact that categories are subsampled to obtain balanced SVM training data across categories, the remainder of data can be used as "held out" data for the stacking process. However, subsampling must also to be implemented for the minority cate-

gory, otherwise the stacking process is unable to generalise for the minority category.

D) Use less temporal context. The applicability of this option depends on how much temporal context is required for best generalisation.

Each of the above options has some drawback. However, recall that as part of SVM training, the data from all categories is subsampled in order to be balanced with the category containing least amount of examples. This makes option C attractive, as the resulting dimension can now be reduced substantially (factor of 3 or better). The result is that a large amount of each category except the minority category is unseen by the training algorithm and can be used in the stacking process for learning the bias of the base classifiers. However, there will not be any examples from the minority category that are unseen, and so the biases for this category cannot be adapted to.

In order to have some unseen examples from the minority category, it can also be subsampled. Instead of obtaining SVM training data from samples from the full training pool $\mathcal{P}^m_{N^m \times P}$ of N^m points with dimension P for each category m, some subset $\hat{\mathcal{P}}^m$ is used. For each category m, first the lower and upper bound vectors, $\underline{L}^m$ and $\underline{U}^m$ respectively, are found;

$$\underline{L}^m_p = \text{2-percentile of } \mathcal{P}^m_{np} \text{ along } n, \tag{4.14}$$

$$\underline{U}^m_p = \text{98-percentile of } \mathcal{P}^m_{np} \text{ along } n\ . \tag{4.15}$$

Then, reduced pools $\hat{\mathcal{P}}^m$ for each category are found as

$$\hat{\mathcal{P}}^m = \mathcal{P}^m_{np},\ \{n \mid \underline{L}^m_p \leq \mathcal{P}^m_{np} \leq \underline{U}^m_p,\ \forall p\}\ . \tag{4.16}$$

Using $\hat{\mathcal{P}}^m$, subsampling to obtain balanced training data for SVMs can then be performed, and some amount of data will be held out from all categories

in order for the stacked generalisation procedure to be effective without cross validation. This has an additional advantage that the base SVMs are faster to train due to fewer training data, and the points used for training are more likely to be "representative" of the categories.

Due to the expected high dimension of the temporally augmented data $\overline{\mathcal{D}}$, used for stacking, only the simple LDA classifier is considered. This is because linear boundaries can be expected to perform just as well as non-linear boundaries in high-dimensional spaces. Additionally, LDA is a very computationally cheap option for the higher level classifier, allowing for the development of many models with different amounts of temporal context in order to assess the impact of its inclusion.

4.3.4 Concatenated features representation

In section 4.3.2, a hierarchical approach to decision making was described. The motivation for such an approach is that some representation spaces may perform better at certain parts of the label hierarchy than others. However, the construction is still somewhat ad-hoc, although utilising prior knowledge on the label hierarchy. In section 4.3.3, a method was described for the joint learning of a temporal aggregation combiner and loss decoding function for multi category SVMs. It is just as conceivable that combining the different representation spaces rather than constructing a hierarchical approach manually can also result in increased performance.

Given the untransformed ECG segments, $\mathcal{U}$, R transformation functions T^r and learner function L, form $\mathcal{D}$ by concatenating the different transformed feature vectors, i.e.,

$$\mathcal{D}_i = \left[T^1\left(\mathcal{U}_i\right), \ldots, T^R\left(\mathcal{U}_i\right)\right], \ \forall i \ . \tag{4.17}$$

Then, $\mathcal{D}$ can be substituted directly into (4.11), (4.12) and (4.13) in order to

perform temporal stacking. This method is simply feature vector concatenation prior to learning the base models.

Because the features values from different representation spaces may have different ranges, the features with smaller value ranges (Heur8) are unlikely to affect the learned function with non-adaptive basis functions such as the RBF kernel. Since Spectra and Heur8 features are both positive valued, the 95-percentile value across the training data for each feature is found and used to form a normalisation vector. Since this process discards energy information from the Spectra representation, a single additional feature is constructed which is simply the energy of the segment after mean subtraction. This is also normalised by its 95-percentile value. The choice of 95-percentile value rather than the maximum is justified by the possibility that some extreme values may be many orders of magnitude larger, effectively forcing the remainder of the values into a much smaller and indistinguishable range. On the other hand, if feature value distributions are not long tailed, this choice makes no difference.

4.3.5 Stacking over multiple feature spaces

Previously, feature concatenation as an alternative to constructing classifier hierarchies for exploiting label taxonomy was proposed. This does not, however, provide a hierarchical structure at the top of the classification chain, i.e. the stacked generalisation classifier. Additionally, the non-adaptive kernel functions used with the base SVM classifiers may not be able to make best use of the extra information. To better represent a hierarchical type construction at the stacking level, multiple data sets, ${}^{r}\mathcal{D}^{n}$, are formed, one for each of the R different transformation functions. First the cross validation folds are formed on untransformed

data, $\hat{\mathcal{U}}^n = \mathcal{U} \setminus \mathcal{U}^n$, then, the transformed cross validation folds are computed;

$$ {}^r\mathcal{D}^n = T^r\left(\mathcal{U}^n\right),\ \forall n \in \{1,\ldots,N\}, r \in \{1,\ldots,R\} \tag{4.18} $$

$$ {}^r\mathcal{D} = \bigcup_n {}^r\mathcal{D}^n\ . \tag{4.19} $$

Then, models are learned for each representation space and cross-validation fold as

$$ f_r^n = L\left({}^r\hat{\mathcal{D}}^n\right),\ \forall n, r\ . \tag{4.20} $$

Finally, $\mathcal{D}'$ is formed by

$$ \mathcal{D}'_i = \left[f_1^1\left({}^1\mathcal{D}_i\right), \ldots, f_1^N\left({}^1\mathcal{D}_i\right), \ldots, f_R^1\left({}^R\mathcal{D}_i\right), \ldots, f_R^N\left({}^R\mathcal{D}_i\right)\right],\ \forall i\ . \tag{4.21} $$

Now, $\mathcal{D}'$ is substituted into (4.13) in order to obtain temporally stacked data for training. Again, N-fold cross validation may be eschewed in favour of training pool reduction, via (4.16).

As with the previous section, the dimension of the final augmented data is an important consideration. In the case of two representation spaces, the dimension of $\overline{\mathcal{D}}$ doubles. Therefore, for the same reasons as mentioned previously, only LDA is considered as the stacking classifier.

4.4 Assessments and analysis

4.4.1 Evaluation procedure

The evaluation procedure utilised for obtaining experimental results is almost identical to that of the preceding chapter (see 3.3), except that 200 bootstrap resamples are used instead of 50, to alleviate the problem observed where boxplot notches were extending below the lower quartile (Figure 3.4). This was indicative

Table 4.1: The experimental parameters and values for experiments conducted in this chapter, with acronyms for referencing methods in tables and figures.

Parameter	Values
Observation length	1 second with overlap, each segment shifted by 0.25 seconds
Temporal context	2 seconds, up to 8 seconds, or 5 segments, up to 29 segments
Classifiers	RBF SVM
Ensembles	LCE, temporal stacked generalisation (SG), temporal stacked generalisation by 3-fold cross validation (SG3CV), hierarchical (H- prefix)
Representations	Spectra, Heur8, Spectra and Heur8 concatenated (S+H8), Spectra and Heur8 trained separately and stacked together (S/H8)

that the number of samples were too few, since the extent of the notches depends on the number of samples. Therefore a larger number of samples allows the notches to remain within the inter-quartile range, and improves the ability to determine whether two medians are significantly different by reducing variation in median estimates. It is expected that 200 samples is robust enough for this purpose.

4.4.2 Ensemble method experiments

Based on the results from Chapter 3, the set of features and classifiers for constructing ensemble methods were reduced, and this is summarised in Table 4.1. There was no distinct advantage of using Heur2 representation space over Heur8 representation space, so only Heur8 was considered for developing ensemble methods with. Additionally, since dimension reduction of the Spectra representation space did not result in any significant gain or loss under the RBF SVM classifier, the reduced representations were omitted from consideration. Since the RBF SVM performed among the best, and most consistently, this was chosen as the base level classifier used for building ensembles with. All base classifiers were developed using 1 s observation segments of ECG, although temporal aggregation was performed over longer periods, up to 8 s. Table 4.2 shows the possible combinations of LCE parameters evaluated.

Table 4.2: The parameters for LCE decoding are listed, and the order in which they are changed. When cycling through parameters, first all variations of temporal context were tested, with other parameters constant. Then, once all temporal context variations were tested, the loss function is changed to its next value, and the temporal context is varied again. Similarly when all loss function variations have been tested, the aggregation function is varied to its next value. This occurs for the parameter order as shown in the first column, with the order of parameter values in the second column. The aggregation function and aggregation level parameters are not relevant for SG type ensembles, but instead "no loss decoding" is a final parameter value for the loss function parameter with SG type ensembles

Parameter	**Values and ordering**
Temporal context	2 seconds, up to 8 seconds, i.e. 5 segments, up to 29 segments
Loss function	hinge, linear, exponential, Hamming
Aggregation function	mean, median, majority vote, maximum absolute value
Aggregation level	Before loss decoding, after loss decoding

Where stacking was used with the cross validation method, the folds of data were formed by patient records rather than observation, since the main cause of variation is due to distinct patients.

4.4.3 Results for temporal ensembles with LCEs and stacking

Experiments with LCE, SG and SG3CV type ensembles were performed for the Heur8, Spectra, and S+H8 representations. A large number of parameter combinations were considered, including the choice of aggregation function, loss decoding function, amount of temporal context, and whether aggregation was performed before or after loss decoding. For each representation space and classifier combination, the Acc_{bal} distributions corresponding to the parameters with the highest median Acc_{bal} score are presented in Figure 4.1. For each of the methods where Acc_{bal} distributions are shown, the sensitivity distributions for each individual category NVR, VT and VF are also shown in Figure 4.2. Table 4.3 shows the parameters for each of the best performing methods.

From Figure 4.1 it can be seen that for Spectra and S+H8 representations, the SG3CV median Acc_{bal} for the best parameters was significantly lower than the

Table 4.3: The best performing parameter combinations are listed for each of the different ensemble types built with each representation space

Representation	Ensemble	Best parameter combination
Spectra	LCE	5 s temporal aggregation, Hamming loss decoding, aggregation with minimum loss after loss decoding
Heur8	LCE	7 s temporal aggregation, exponential loss decoding, aggregation with mean loss after loss decoding
S+H8	LCE	8 s temporal aggregation, Hamming loss decoding, aggregation with median function before loss decoding
Spectra	SG	8 s temporal aggregation, stacking Hamming loss decoding values
Heur8	SG	8 s temporal aggregation, stacking without loss decoding
S+H8	SG	8 s temporal aggregation, stacking linear loss decoding values
Spectra	SG3CV	7 s temporal aggregation, stacking without loss decoding
Heur8	SG3CV	8 s temporal aggregation, stacking hinge loss decoding values
S+H8	SG3CV	7 s temporal aggregation, stacking without loss decoding

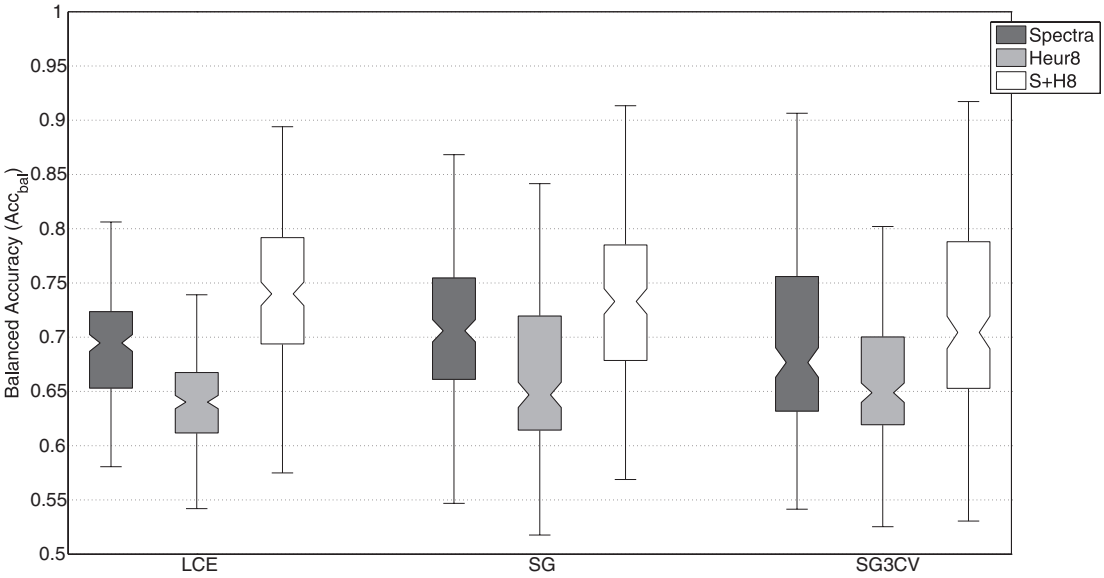


Figure 4.1: Distributions of Acc_{bal} across all bootstrap resamples for LCE, SG and SG3CV ensembles with Spectra, Heur8 and S+H8 representations. In each case these are shown for the best parameter combination by median Acc_{bal} as described by Table 4.3

Table 4.4: Average confusion matrices over all bootstrap resamples for each method shown in Figure 4.1. Rows are the ground truths, and columns are the diagnoses made.

		Diagnosed as		
Method	**Ground Truth**	**NVR%**	**VT%**	**VF%**
LCE Spectra	NVR	82.5	6.2	11.3
	VT	21.0	53.7	25.3
	VF	6.2	22.8	71.0
LCE Heur8	NVR	93.7	4.2	2.2
	VT	17.9	18.1	64.0
	VF	4.0	12.5	83.4
LCE S+H8	NVR	89.8	5.2	5.0
	VT	18.4	51.9	29.7
	VF	3.7	15.6	80.7
SG Spectra	NVR	83.9	5.3	10.8
	VT	22.4	49.1	28.5
	VF	3.1	17.6	79.2
SG Heur8	NVR	95.8	3.0	1.2
	VT	20.6	26.7	52.7
	VF	4.4	17.3	78.3
SG S+H8	NVR	90.5	4.6	4.9
	VT	17.9	52.6	29.6
	VF	2.5	20.3	77.2
SG3CV Spectra	NVR	91.1	4.2	4.7
	VT	24.1	41.7	34.2
	VF	11.2	13.5	75.3
SG3CV Heur8	NVR	97.5	1.5	1.0
	VT	22.4	27.1	50.4
	VF	6.3	17.5	76.2
SG3CV S+H8	NVR	96.5	1.9	1.6
	VT	28.6	39.8	31.6
	VF	8.2	12.4	79.4

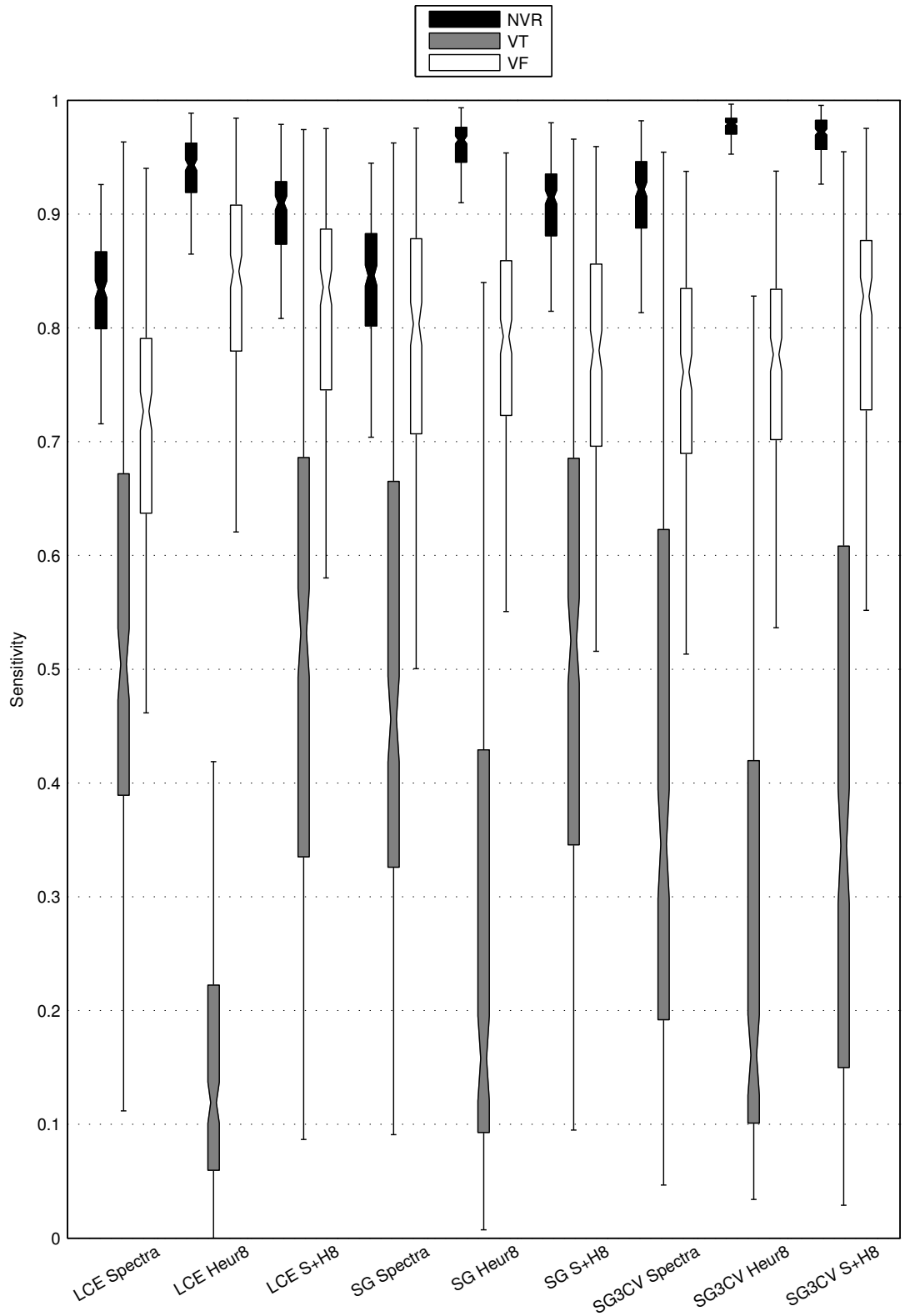


Figure 4.2: Distributions of sensitivities for each of NVR, VT and VF for the Spectra, Heur8 and S+H8 representations classified using LCE, SG and SG3CV ensemble methods, for their corresponding best parameters as described by Table 4.3

SG median Acc_{bal} for the best parameters. For the Heur8 representation, there was no difference, however, it can be seen from Figure 4.2 that NVR sensitivity of Heur8 SG3CV was significantly higher than the Heur8 SG method. Therefore, further results for SG3CV ensembles are not presented, except for construction of hierarchical classifiers using Heur8 SG3CV.

In order to gain some intuition about the impact of the selection of loss function (where applicable), aggregation function, and amount of temporal context on the classification accuracy, surface plots of median Acc_{bal} scores for each representation with all the possible different parameter combinations for LCEs and SG classifiers are shown in Figure 4.3. These figures show that selection of appropriate aggregation options was important for both LCE and SG ensembles, and also that increasing temporal context provided a small improvement in classification accuracy.

According to confusion matrices shown in Table 4.4 with all ensemble types, Heur8 representation space had the best sensitivity for NVR rhythms, while reducing the amount of false NVR assignments from VT or VF (see Table 3.2). The level of incorrect assignments to NVR from VT and VF was lower for S+H8 classified using SG, however the NVR sensitivity was substantially lower. Therefore, when considering hierarchical constructions, only the Heur8 based classifiers were considered as the master classifier.

4.4.4 Results for hierarchical and stacked hierarchical constructions

Figure 4.4 shows the Acc_{bal} distributions for all the hierarchical constructions considered, and Figure 4.5 shows the sensitivity of each category NVR, VT and VF for the same hierarchical constructions. Finally, each construction is summarised by Table 4.5, showing average confusion matrices.

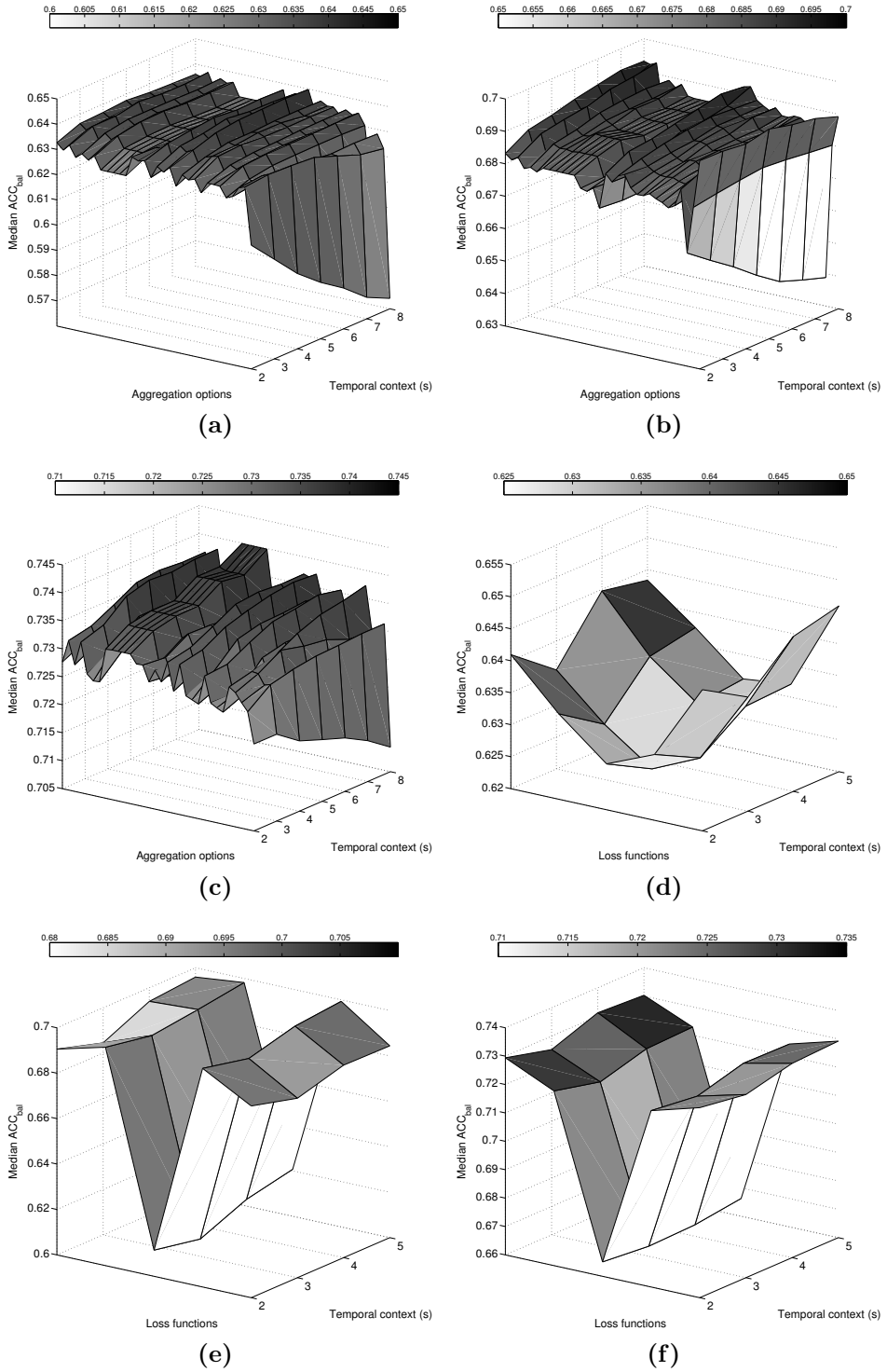


Figure 4.3: Surface plots showing the median Acc_{bal} scores for the different SG and LCE decoding parameters varied, in the order described by Table 4.2, and with the amount of temporal context split into a separate axis. The results are shown for LCEs with; (a) Heur8 representation, (b) Spectra representation, and (c) S+H8 representation, and for SGs with; (d) Heur8 representation, (e) Spectra representation, and (f) S+H8 representation

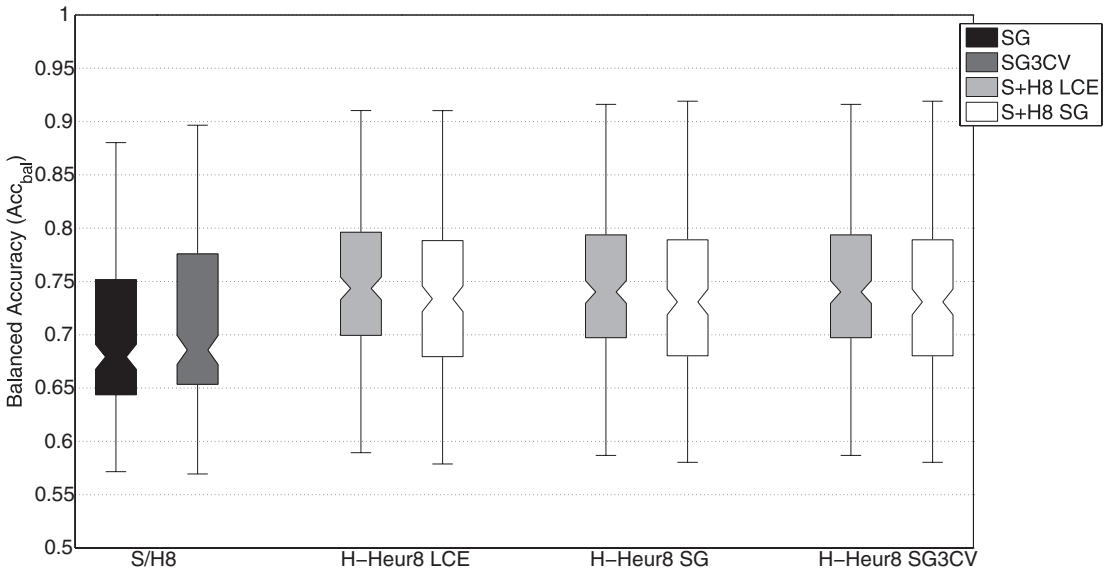


Figure 4.4: Distributions of Acc_{bal} across all bootstrap resamples for hierarchical constructions formed by S/H8 representation classified using SG and SG3CV methods, and Heur8 master classifiers with LCE, SG or SG3CV methods with secondary decisions made using S+H8 with either LCE or SG methods

Table 4.5: Average confusion matrices over all bootstrap resamples for each method shown in Figure 4.4. Rows are the ground truths, and columns are the diagnoses made.

		Diagnosed as		
Method	**Ground Truth**	**NVR%**	**VT%**	**VF%**
S/H8 SG	NVR	96.6	1.9	1.5
	VT	21.0	33.9	45.1
	VF	6.2	13.1	80.7
S/H8 SG3CV	NVR	97.7	1.1	1.2
	VT	23.4	35.2	41.5
	VF	7.9	11.6	80.5
H-LCE Heur8 LCE S+H8	NVR	94.4	2.6	3.0
	VT	20.4	50.0	29.6
	VF	4.9	15.3	79.8
H-LCE Heur8 SG S+H8	NVR	95.1	2.1	2.8
	VT	20.9	50.0	29.1
	VF	5.0	19.9	75.1
H-SG Heur8 LCE S+H8	NVR	96.4	1.6	2.0
	VT	22.9	47.8	29.3
	VF	6.0	15.0	79.0
H-SG Heur8 SG S+H8	NVR	96.2	1.6	2.2
	VT	22.3	48.9	28.8
	VF	4.8	19.6	75.5
H-SG3CV Heur8 LCE S+H8	NVR	96.4	1.6	2.0
	VT	22.9	47.8	29.3
	VF	6.0	15.0	79.0
H-SG3CV Heur8 SG S+H8	NVR	96.2	1.6	2.2
	VT	22.3	48.9	28.8
	VF	4.8	19.6	75.5

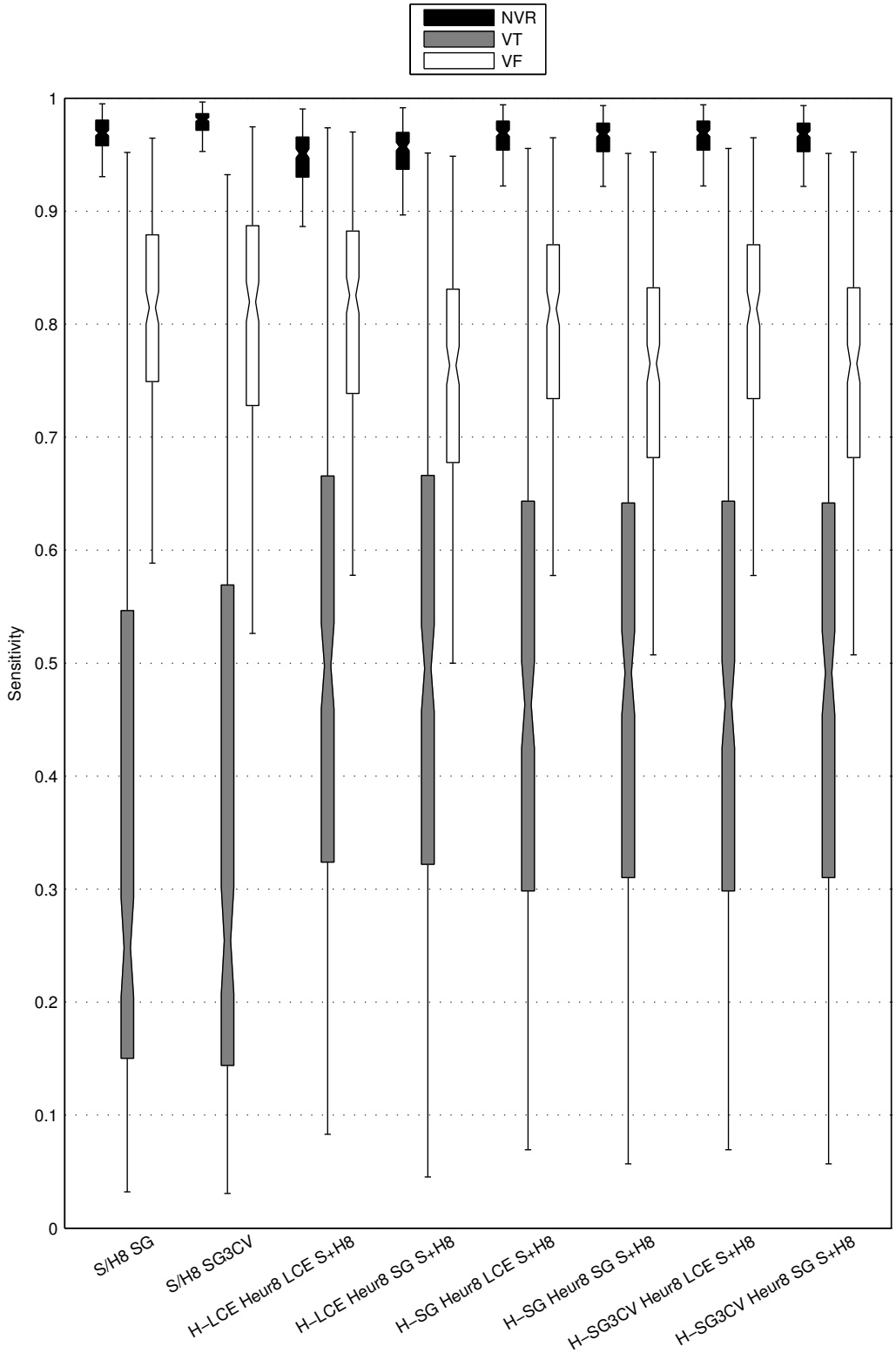


Figure 4.5: Distributions of sensitivities for each of NVR, VT and VF across all bootstrap resamples of hierarchical constructions formed by S/H8 representation classified using SG and SG3CV methods, and Heur8 hierarchical master classifiers with LCE, SG or SG3CV ensembles and secondary decisions made using S+H8 classified with LCE or SG methods

The results presented previously demonstrated that for making a NVR vs arrhythmia decision, the Heur8 representation obtained significantly better NVR sensitivity than other representations with all ensemble methods. However, the S+H8 representation obtained higher correct rejection rate of NVR, and also significantly higher VT sensitivity. Hierarchical classifiers were therefore composed of a master classifier based on Heur8 representation space, with either LCE, SG or SG3CV ensembles, and a secondary classifier based on S+H8 representation with either LCE or SG ensembles. Additionally, a less ad-hoc hierarchy was constructed via stacking of SVMs trained on Spectra and Heur8 representations separately, as described in 4.3.5 with both SG and SG3CV type ensembles.

4.5 Discussion

A large number of experiments were performed, due to substantial combinations of selectable parameters for the various types of ensemble methods investigated. For LCEs alone, 224 distinct combinations of parameters were possible, and with SG type ensembles, 35 distinct combinations of parameter were possible. The median Acc_{bal} scores for these ensemble methods combined with each investigated representation space was visualised as surface plots in Figure 4.3. This showed that increasing amounts of temporal context improved the median Acc_{bal} scores somewhat, and also that the choice of other combiners (loss function where applicable, aggregation function where applicable) had a substantial impact on the overall performance of the ensemble, with the exponential loss function in particular performing poorly with stacking type ensembles. Notably, the cross-validated type of stacked ensemble generally performed either no better, or worse, than the stacked ensemble without cross validation.

All classification ensembles were built using 1 s segments, despite this not being the best choice for Heur8 representation, according to results shown in

3.4.2. Both Spectra and Heur8 representations gained $2-3\%$ median Acc_{bal} scores by considering LCEs formed over SVM decision values or loss values, shown in Figure 4.1. For ensembles formed by stacked generalisation there was also an additional improvement over LCEs, with improvement to the upper quartile value of about 6% for Heur8 representation. An important aspect of improvement for the Heur8 representation was that, according to confusion matrices in Table 4.4, the NVR sensitivity was improved, whilst also reducing false NVR diagnoses. This appeared to be somewhat of a trade-off, with increasing NVR sensitivity corresponding with an increase in false NVR diagnoses. On the other hand, Figure 4.2 shows that the temporal ensemble methods improved the sensitivity of NVR in some cases, and also improved VT and VF sensitivities in other cases, although with a trade-off, i.e. increases in sensitivity of a given category corresponds with decreases in one or more of the other categories.

In order to try and minimise the impact of the tradeoff, three approaches for combining predictive power of different representations were tested. The representations could be concatenated at the feature vector level, for training of the base SVMs. Additionally, hierarchical approaches were considered based on an obvious hierarchy in the label taxonomy. These were constructed directly, by replacing decisions from one classifier with another, or indirectly by building a stacked generalisation process on top of classifiers trained in different representation spaces. The results for feature vector concatenation were shown in Figure 4.1 and Table 4.4, alongside results for temporal ensembles constructed for Spectra and Heur8 representations, and shown to give substantial improvements, particularly to VT sensitivity without degrading VF sensitivity. This motivated the use of S+H8 representation as the secondary classifier for hierarchical constructions, rather than Spectra based classifiers. The amount of possible hierarchical constructions was very large ($(224+35)^2$ possible combinations), and were not explored in entirety. These were instead constructed by combining the best LCE

and SG parameters for Heur8 (master classifier) and S+H8 (secondary classifier) representations. In addition to this, Spectra and Heur8 classifiers were trained separately and the SVM outputs from both classifiers stacked temporally. It was shown in Figure 4.4 that stacking classifiers trained on separate Spectra and Heur8 representations performed the least well out of the hierarchical constructions, however, according to Figure 4.5 and Table 4.5, NVR sensitivity was substantially higher, but at the cost of substantially lower VT sensitivities. On the other hand, hierarchies constructed via decision replacements all performed roughly similarly according to Figure 4.4, however it was seen from Figure 4.5 that this was the result of small tradeoffs between NVR, VT and VF sensitivities.

4.5.1 Evolution of contributions and the best result

In order to see the impact of each contributions, Figure 4.6 shows Acc_{bal} distributions for Heur8 and Spectra representations with non-overlapping segments of length 8 s and 1 s respectively, and some of the best performing temporal stacking and hierarchical decoding combinations. The parameters presented for Heur8 are those as shown in the original literature, i.e. classification with the RBF kernel, non overlapping 8 s segments. For these same methods, Figure 4.7 shows the sensitivities of each rhythm. It can be seen that the interquartile ranges were not reduced, however, substantial improvement in the median Acc_{bal} (5%) was achieved by considering temporal stacking of the Spectra representation space, due to improvements in both median NVR sensitivity and median VT sensitivity without any reduction of median VF sensitivity when compared to Spectra classified without any temporal stacking. Further improvements upon this were obtained by various methods of combining Heur8 and Spectra representations. In the simplest form, concatenating these representations and using LCEs resulted in 3% improvement in median Acc_{bal} over classification of Spectra

representation using stacked temporal ensembles. The median NVR sensitivity improved to 91%, 2% less than that for the benchmark Heur8 method classified with non-overlapping segments. However, median sensitivities for VT and VF were substantially higher, 53% vs 18%, and 84% vs 74%, respectively. Further improvement was obtained by a hierarchical decision structure where a Heur8 classifier first decided if NVR was present or not, and if not, a final decision was taken by a classifier built using concatenated Spectra and Heur8 features. The resulting median sensitivities for NVR, VT and VF became 95%, 50% and 83%, respectively.

Other hierarchical constructions shown, demonstrated that improvements in sensitivity for a given category were traded off for reduced sensitivity of another category. For example, temporal stacking of Spectra and Heur8 simultaneously resulted in the highest median NVR sensitivity of 98%, but median VT sensitivity dropped to 25%, whilst VF sensitivity remained similar, at 82%.

The very best classification method presented in this chapter is a complex combination of hierarchical and temporal methods. The classifier was constructed by checking the decision of a 7 s LCE classifier trained over Heur8 features, which if was any other than NVR, taking as the final decision the output of an 8 s LCE classifier trained over the Spectra features. As mentioned before, for testing hierarchical classifiers there were $(224 + 35)^2$ possible combinations, most of which were unexplored. The combinations tested were instead simply combinations of the best performing methods from each category. It is conceivable that some small gain is possible by exploring this space in its entirety. In addition, the lengths over which the temporal aggregations are performed are substantial, and not any shorter than window lengths selected by previous methods. The best classifier may be improved further by considering a NVR vs ventricular rhythms binary classifier trained with Heur8, temporally aggregated over a 5 s to 8 s period, and with secondary decisions made by another classifier trained

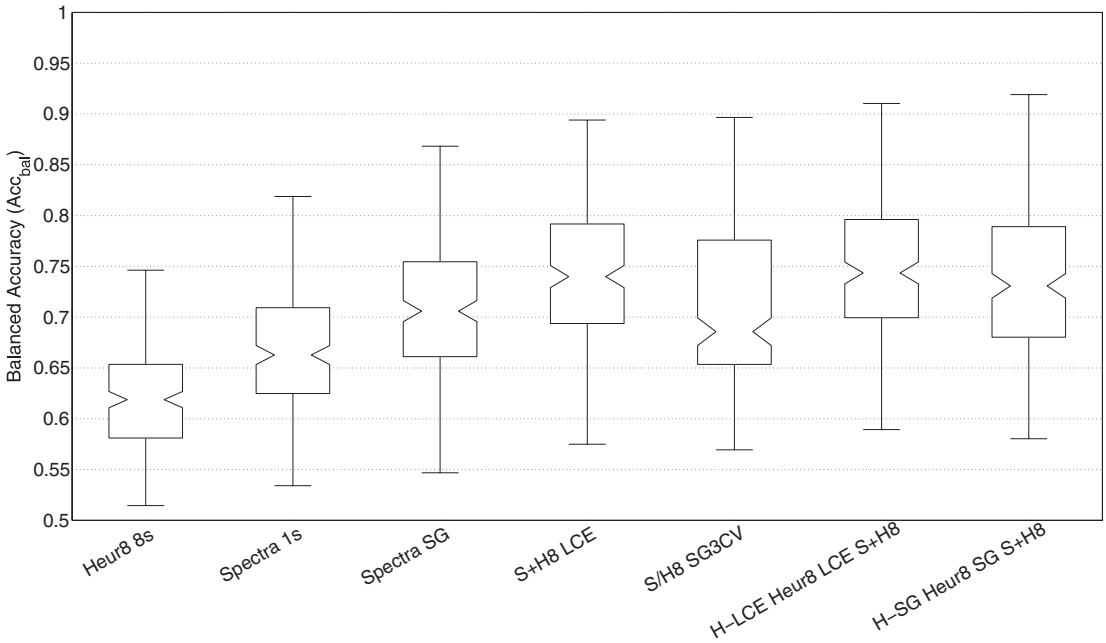


Figure 4.6: Distributions of Acc_{bal} across all bootstrap resamples for Heur8 and Spectra reference classifiers, the best performing Spectra temporal ensemble, the best performing concatenated features temporal ensemble, hierarchical classification via stacking Spectra and Heur8 separately, and hierarchical decision combining

of S+H8, aggregated over a shorter period, e.g. 2 s. The motivation for this would be to allow the primary classifier to be focussed where it performs best – differentiation between ventricular and non-ventricular rhythms. The use of a shorter temporal aggregation length over the secondary classifier would be to allow better distinction of transient ventricular arrhythmias. However the gains expected from these optimisations are likely to be small. Results obtained with the best classifier, were based on feature concatenation and hierarchical decision making based on larger sets of features than considered in the state of the art. It seems likely that effort to derive better features, particularly for discriminating between VT and VF are most likely to result in substantial further improvements in classification accuracy.

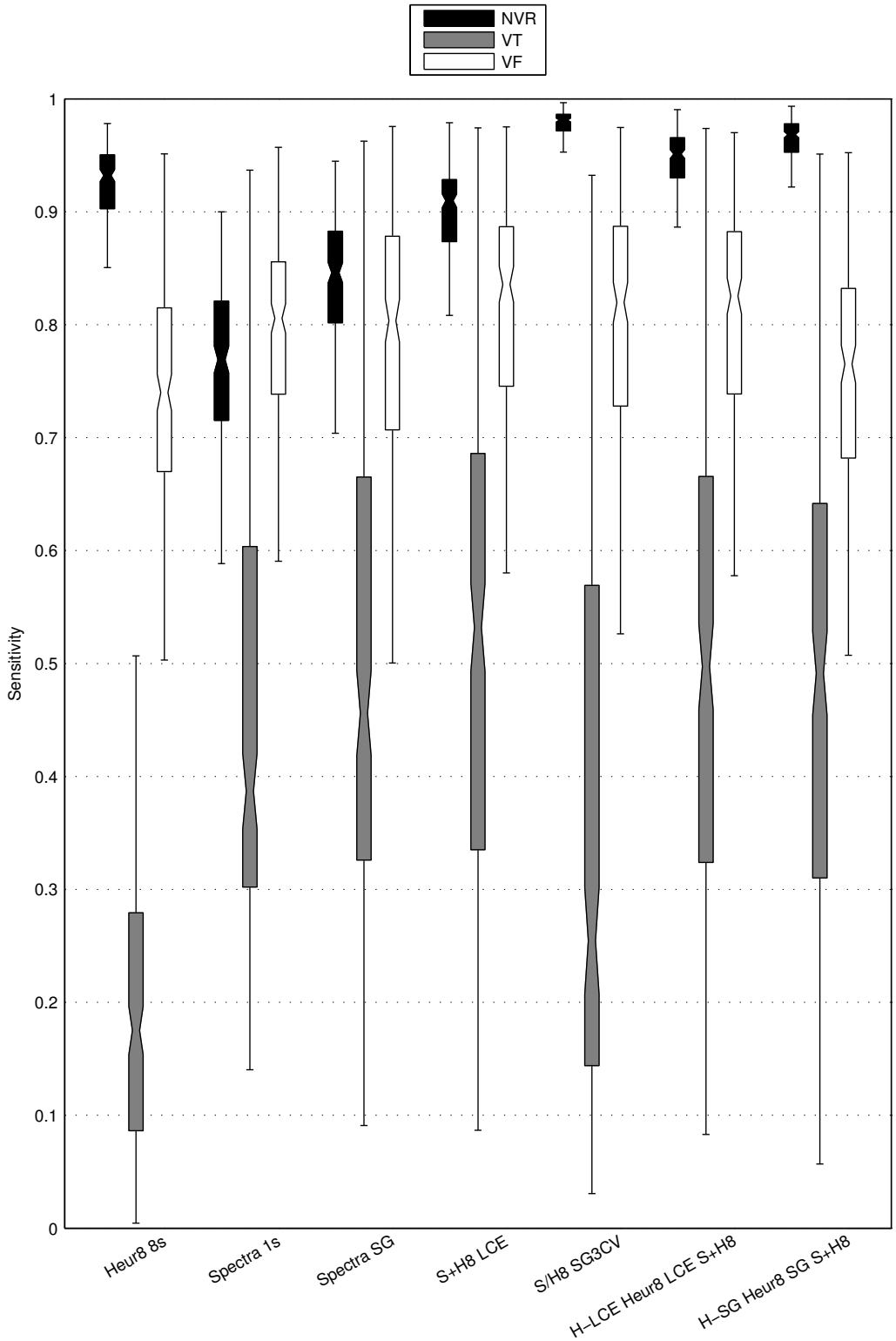


Figure 4.7: Distributions of per category sensitivities across all bootstrap resamples for Heur8 and Spectra reference classifiers, the best performing Spectra temporal ensemble, the best performing concatenated features temporal ensemble, hierarchical classification via stacking Spectra and Heur8 separately, and hierarchical decision combining

4.5.2 Clinical importance of results

The problem at hand is to produce a classification algorithm which performs better at differentiating between VF and VT whilst not introducing false alarms or failing to identify either of these rhythms. It was determined experimentally that previous methods were capable to distinguish between normal rhythms and life threatening rhythms of a ventricular origin. However, they were unable to correctly tell apart VT and VF, classifying a majority of ventricular rhythms as VF. From confusion matrices in Chapter 3, Table 3.2, it can be seen that the Heur8 method as originally posed missed VT on average 13% of the time, and VF 8% of the time. On the other hand, a false alarm (where the rhythm was normal) was raised for VT 5.5% of the time while a false alarm (where the rhythm was normal) was raised 2.4% for VF. This means that an inappropriate treatment might be delivered for up to 8% of normal rhythms, while 8% of VF instances are missed for treatment, and VT is missed for any form of treatment in 13% of cases.

On the other hand, as presented in confusion matrices (Table 4.5), the best hierarchical classifier missed VF for treatment in only 5% of cases. However, the VF false alarm rate was raised by 0.6%. VT was missed for any form of treatment in more cases, 20%, but the amount of VT false alarms was reduced to 2.6% of normal rhythms. However, the Heur8 classifier misidentifies VF as VT 21% of the time, which would result in an ineffective cardioversion treatment, while the hierarchical classifier reduces this rate to 15%. As well as this, the Heur8 classifier would deliver a too intense defibrillation shock in 55% of VT instances, while the hierarchical classifier again improves upon this, reducing the rate of VT falsely diagnosed as VF to 30% of instances. From this perspective it is clear that in almost all regard the hierarchical classifier performed better, at the expense of missing VT for any form of treatment more often. Overall,

the hierarchical classifier improved upon state of the art, but the false alarm rate was still too high, and VF was still missed in 5% of cases. Since the diagnoses were given every 0.25 s, it might be possible to refine the decision making rules to provide highly accurate treatment delivery, for example, if VF is indicated 9 times in a row (over a 2 s period), then a defibrillation shock can be administered. The final problem remains however, of improving the ability to tell between VF and VT. While it is clear that the hierarchical method has improved upon the detection accuracy of these categories, at 50% and 83% respectively (Figure 4.5), the result is still clearly not satisfactory in terms of differentiating between VT and VF, and the algorithm presented is not suitable for use in a clinical setting.

4.6 Summary and conclusions

Previous approaches to making diagnoses in the ECG relied upon adopting a window based approach, by splitting the ECG into non-overlapping segments, performing feature transformation on the segment and then classifying the derived features. The drawbacks of such an approach was that the decisions were not updated frequently, and the use of longer segments was often mandated to ensure key observations in the ECG were not present only at the boundaries of the segments, e.g. QRS complexes appearing at the start of a short ECG segment. Another effect of this type of construction was that the same low dimensional features were derived over long or short observations, which fails to capture any kind of temporal evolution which may be useful for discrimination between rhythms.

Therefore a key proposal of this chapter was to form ensembles over SVM outputs using short, overlapping segments of ECG. This was implemented by capturing temporal evolution after evaluating scores by either modifying the decision process to perform some temporal aggregation of scores, or using a

classifier stacking process to automatically weight the impact of previous scores. The result was a substantial improvement in overall accuracy, as well as an improvement in sensitivity of all categories.

Despite these improvements, there were still very clear differences in the discrimination ability between Heur8 and Spectra representations, in particular, Spectra representation NVR sensitivity was still unacceptably low. In order to obtain all around improvements in sensitivities of all categories simultaneously, three methods were evaluated for combining the predictive power of the different representation spaces. The first was to simply concatenate feature spaces, and construct temporal ensembles from SVMs trained using these concatenated features. The second was to construct a hierarchical taxonomy for classification, and delegate responsibility for certain diagnoses to different classifiers. Finally, rather than ad-hoc constructions, classifier stacking was used directly by combining temporal ensembles of each representation space considered separately in an attempt to exploit category specific strengths automatically. The most substantial improvement was obtained by concatenating feature spaces for training the base SVMs in combination with temporal stacking. This brought the NVR sensitivity almost up to par with the Heur8 benchmark, while providing substantial improvements to VT and VF sensitivities simultaneously. The hierarchical constructions then allowed building of classifiers which exceeded the benchmark NVR sensitivity, with minimal reduction of VT and VF sensitivities.

Chapter 5

Conditional random fields for sequential ECG labelling

In the previous chapter, methods were elaborated to construct classifiers which used temporally local classifier outputs in order to enhance diagnostic performance, exploiting the fact that the ECG is a time series. However, the methods developed were initially ad-hoc, utilising preselected aggregation functions in order to temporally aggregate output scores before decoding, or temporally aggregate decoded loss values. More integral versions of the LCE procedure were developed via stacked generalisation, however, there exist a class of methods known as *structured prediction* which are intended to exploit interactions between neighbouring, or even long range observations and outputs. A well known and basic method for sequence modelling is the hidden Markov model. A variation of this model is the conditional random field, which for a linear chain structure is the discriminative equivalent of a hidden Markov model. Since conditional random fields are developed as natural solutions to sequence labelling tasks, it is useful to study their applicability to ECG rhythm labelling.

5.1 Chapter outline

In this chapter, a brief descriptive overview of sequential supervised learning is given. Then, one of the most basic and popular methods for sequential learning, hidden Markov models, is described, and some of their limitations are discussed. Then discriminative methods for sequential labelling, maximum entropy Markov models and conditional random fields, are described briefly, and how they address the limitations of hidden Markov models. Finally,brief experimentation using conditional random fields is performed. The methods and experimental results are described, and options for future experimental work utilising these methods are explored, given sufficient time.

5.2 Structured prediction overview

Usually, a classifier takes a vector as input and produces a decision about which category the vector belongs to. However, the vector is considered in isolation, and if the vectors are presented in a sequence, neighbouring vectors in the sequence have no impact on the outcome. For time series prediction, and other sequential data, it is unlikely to be the case that nearby parts of the observation sequences are uncorrelated or independent of the current observation, so approaches which consider nearby observations are potentially useful.

This is the motivation behind methods for structured prediction, or supervised sequential labelling. A review of sequential learning [72] lists some methods, including sliding windows, recurrent sliding windows, recurrent neural networks, hidden Markov models (HMMs) and conditional random fields (CRFs). It turns out, that the temporal ensembles method employed in the previous chapter is similar, but not identical, to the recurrent sliding window approach [73]. Methods for structured prediction find use in many application domains [74], including but

not limited to; automatic speech recognition, activity and gesture recognition, part of speech tagging, gene discovery and protein sequence alignment.

The most widely used method, HMMs, are discussed, including their limitations and resulting evolution to CRFs.

5.2.1 Generative sequence modelling: hidden Markov models

In order to model data sequentially, a sequence of states $\mathbf{s}$ is assumed to generate a sequence of observations $\mathbf{o}$. The states are discrete, and the observations are commonly discrete, but may also be continuous through the use of a Gaussian mixture model for observations, although there still need to be a finite number of generating Gaussians, or otherwise some form of vector quantisation is required to map continuous values into discrete observations. A HMM models the joint probability density of sequences and observations, $p(\mathbf{o}, \mathbf{s})$, but in order for this modelling to be feasible the following simplifying assumptions are made;

1. Conditional independence of the sequence of states, i.e. states are conditionally independent of all other states given only the previous state (otherwise known as the Markov property)

2. Observation are conditionally independent of all other observations given the state that generated it.

These assumptions, while making HMM parameter estimation feasible, also introduce some limitations. Conditional independence of the state sequence means that long distance interactions are not modelled, and conditional independence of the observation sequences means that past observations are not used for inferring

the current state. In particular, this means a HMM model is of the form [75]

$$p(\mathbf{o}, \mathbf{s}) = \prod_t p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{o}_t|\mathbf{s}_t) \ . \tag{5.1}$$

Despite the fact that such a modelling fits the ECG well, i.e. there is an underlying state (NVR, VT, VF) generating an observation, modelling this directly is not necessarily the most efficient way to obtain diagnoses of the underlying state since what is actually required is $p(\mathbf{s}|\mathbf{o})$. This can be found easily using Bayes' theorem, however, if all that is required is $p(\mathbf{s}|\mathbf{o})$ then resources are wasted by modelling $p(\mathbf{o}, \mathbf{s})$. In particular, the estimates of $p(\mathbf{s}|\mathbf{o})$ formed from $p(\mathbf{o}, \mathbf{s})$ may not be as accurate as those formed directly with other methods.

Training of a HMM is easy if the class labels are taken to be the hidden states – the hidden state transition probabilities and state observation probabilities are simply normalised histogram counts (assuming discretised observations). The model may even be improved using unlabelled data by using these estimated probabilities as initial guesses and using the Baum-Welch training procedure to refine the probability estimates. Then using these learned distributions, given an observation sequence, the most likely state sequence can be computed via the Viterbi algorithm. However using this approach, long range interactions cannot be mediated via the hidden states, as these are assumed to be the class labels. A variety of algorithms exist for inferring the hidden state sequence, including Viterbi, forwards-backwards or filtering [74]. Only the filtering algorithm is suitable for online (real time) label estimation, as the other two require all of the observations be available for inference.

Alternatively, if the hidden states are desired to mediate some long range interactions in the sequence data, then the training segments can be broken into continuous segments of the same label, and these segments used to train a single HMM for each label. Then, to obtain a label for a given sequence, the

probability of the most likely state sequence is found given each HMM, and the label corresponding to the HMM obtaining the highest likelihood is the selected label. However, this approach is only capable of generating a single label per observation sequence, if continuous labelling is required, the sequence needs to be segmented and each segment tested in this fashion. However, using this method means that online labels are always available.

5.2.2 Discriminative sequence modelling: maximum entropy Markov models

From the previous description it can be seen that some deficiencies exist with HMMs that can result in suboptimal performance. The requirement of a finite observation alphabet or set of generators is such a limitation which precludes the use of richer overlapping features, and as seen from the previous chapter, the feature set heavily influences the best performance attainable. Additionally, a HMM is trained to maximise the likelihood that the model generated the training sequences, however in the predictive task of finding $p(\mathbf{s}|\mathbf{o})$, the observations are given and thus there is no need to estimate the probability of an observation. This deficiency is particularly manifest in the segmented type HMMs where the hidden states are used to mediate long range interactions; because the HMMs are trained per category, they are not optimised for maximising separation between categories.

A maximum entropy Markov model (MEMM) proposes to fix some of these deficiencies, by modelling $p(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{o}_t)$ directly [76]. This has the distinction that modelling now takes into account the previous state, therefore the observations are considered as being associated with state transitions rather than the current state only. By avoiding modelling the joint distribution $p(\mathbf{o}, \mathbf{s})$, observations are no longer constrained in their representation, and may be sequences of real valued

vectors. As with HMMs, a forwards-backwards procedure is used for inferring the sequence labels given a sequence of observations and a trained MEMM. However, there exists a limitation of MEMMs, known as label bias. Put simply, the directed nature of the graph underlying the model means that evidence further along the sequence cannot flow backwards to previous states; this is a consequence of conservation of probability mass at each state transition. CRFs were developed to alleviate this specific problem [77].

5.2.3 Discriminative sequence modelling: conditional random fields

As mentioned, the principal shortcoming of MEMMs is the label bias problem, which requires probability mass to be conserved at each state transition, with the result of favouring states with fewer outgoing transitions, irrespective of the observations. The CRFs framework solves this in a principled fashion [77], by allowing label transition probabilities to be globally normalised given the entire sequence, rather than locally. A downside of this approach is that CRFs are not suitable for real time label inference [74, 75]. CRFs do not employ the notion of hidden states in the way a HMM does, instead the framework is developed considering label sequences.

There are other benefits to the CRF approach, as well as solving the label bias problem. They permit arbitrary underlying graphical structures [75, 78, 79], which means that the label sequence no longer requires the Markov assumption, although Markov like linear chain CRFs are still frequently used. For linear chain CRFs, they can be constructed such that label transitions are conditioned on just the current observation, or a number of past and future observations, or even the entire observation sequence. CRFs also have latent variable extensions for mediating complex non-linear and long range interactions between observations and

states, via hidden state CRFs [80], which assigns a single label per sequence requiring sequence segmentation for label sequence predictions. Additionally, latent dynamic CRFs contain latent variables for assigning a label to each observation in a sequence without using sequence segmentation that would be required when using the hidden state CRF [81]. In particular, a modified hidden state CRF is found to be the best performing single-system method in an automatic speech recognition task [82].

5.3 Experimental methods and results

Due to time constraints, it was not possible to investigate the use of structured prediction methods more fully. The methods considered here were preliminary investigations that tried to determine if a simple application of CRFs can improve upon results obtained previously, due to being a purpose designed method for sequential labelling.

In particular, problems with attempting to assess HMMs arise from the need to estimate the number of mixture components (for continuous observations), or number of vector quantisation centres, which would require searching a parameter range and using cross validation. For latent variable CRFs, the training time is considerable, and again the free parameters, amount of observations to condition labels on and number of latent variables, need to be selected by cross validation. Even for standard CRFs without latent variables, training takes sufficiently long that it was not feasible to investigate the impact of number of observations the labels are conditioned upon. Finally, only linear chain CRF structures were considered, due to the time required for engineering feature functions and alternative structure that might be capable of capturing nonlinear label interactions.

For training CRFs with the HCRF software [81], there were some parameters that may be optimised, including a regularisation parameter, and the number of

Table 5.1: The experimental parameters and values for experiments conducted in this chapter.

Parameter	Values
Observation length	1 second with overlap, each segment shifted by 0.25 seconds
Temporal context	25 segments, equivalent to 7 seconds. By default of the HCRF package, this is future observations and past observations with respect to the current label, i.e. 12 past observations, the current observation, and 12 future observations
Classifiers	CRFs trained over SVM outputs, CRFs trained on feature sequences directly
Representations	Heur8, Spectra and Heur8 concatenated (S+H8)
CRF regularisation	0
CRF solver iterations	100

iterations for the optimisation solver. These were simply selected a priori and fixed, as it was not feasible to optimise using cross-validation.

5.3.1 Experimental methods

As with the previous chapter, 200 bootstrap resamples were used for experimental assessments, using the same record randomisation as the results from the previous chapter, to allow them to be comparable. Table 5.1 shows all of the experimental parameters and the values or value ranges they were fixed to.

5.3.2 Results of CRF experiments

Figure 5.1 shows the Acc_{bal} distributions for CRF classifiers trained over both raw features and SVM outputs, with both Heur8 and S+H8 features. Spectra features were not tested since in the previous chapter there was no case where Spectra only features had an advantage over S+H8 features. Then, Figure 5.2 shows the sensitivity of each category NVR, VT and VF for these methods. Finally, each method is summarised by Table 4.5, showing average confusion matrices.

The best performing method, CRFs trained over S+H8 features directly, did not perform much better on average than a Spectra based RBF SVM classifier, with a median Acc_{bal} of 68%. CRFs trained directly over Heur8 features obtained the worst result, with a median Acc_{bal} of 60%. From Figure 5.2 it can be seen

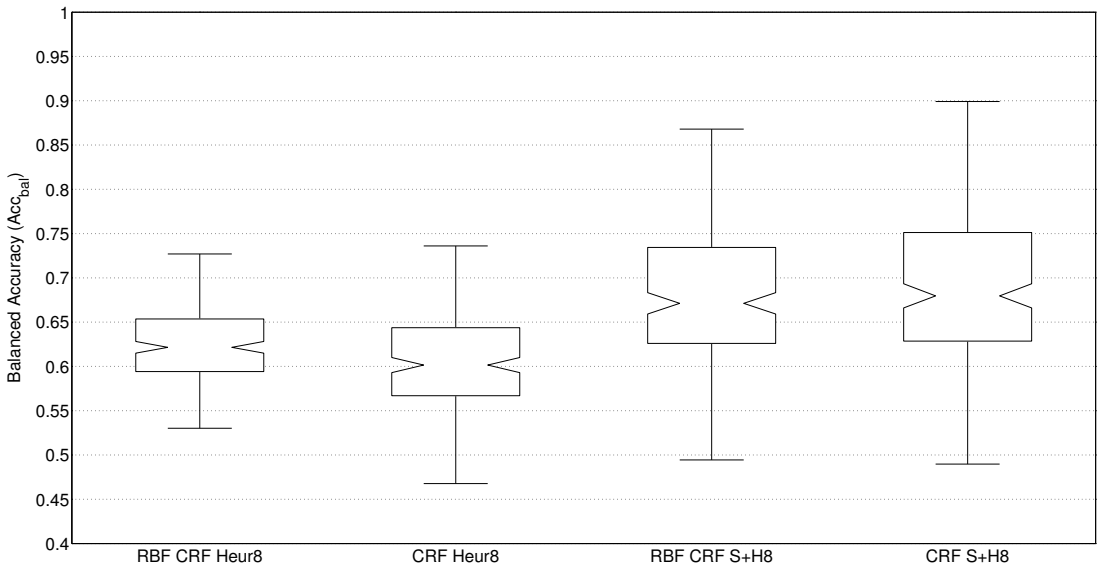


Figure 5.1: Distributions of Acc_{bal} across all bootstrap resamples for CRFs trained over RBF SVM outputs, and trained directly over Heur8 and S+H8 representation spaces

Table 5.2: Average confusion matrices over all bootstrap resamples for each method shown in Figure 5.1. Rows are the ground truths, and columns are the diagnoses made.

		Diagnosed as		
Method	**Ground Truth**	**NVR%**	**VT%**	**VF%**
RBF CRF Heur8	NVR	93.5	3.5	3.0
	VT	25.3	19.6	55.1
	VF	8.2	14.0	77.8
CRF Heur8	NVR	93.0	3.9	3.2
	VT	25.6	19.7	54.7
	VF	10.0	19.4	70.6
RBF CRF S+H8	NVR	94.1	4.0	2.0
	VT	30.1	32.1	37.8
	VF	11.5	10.1	78.4
CRF S+H8	NVR	90.0	6.8	3.2
	VT	23.9	40.9	35.2
	VF	10.5	13.4	76.2

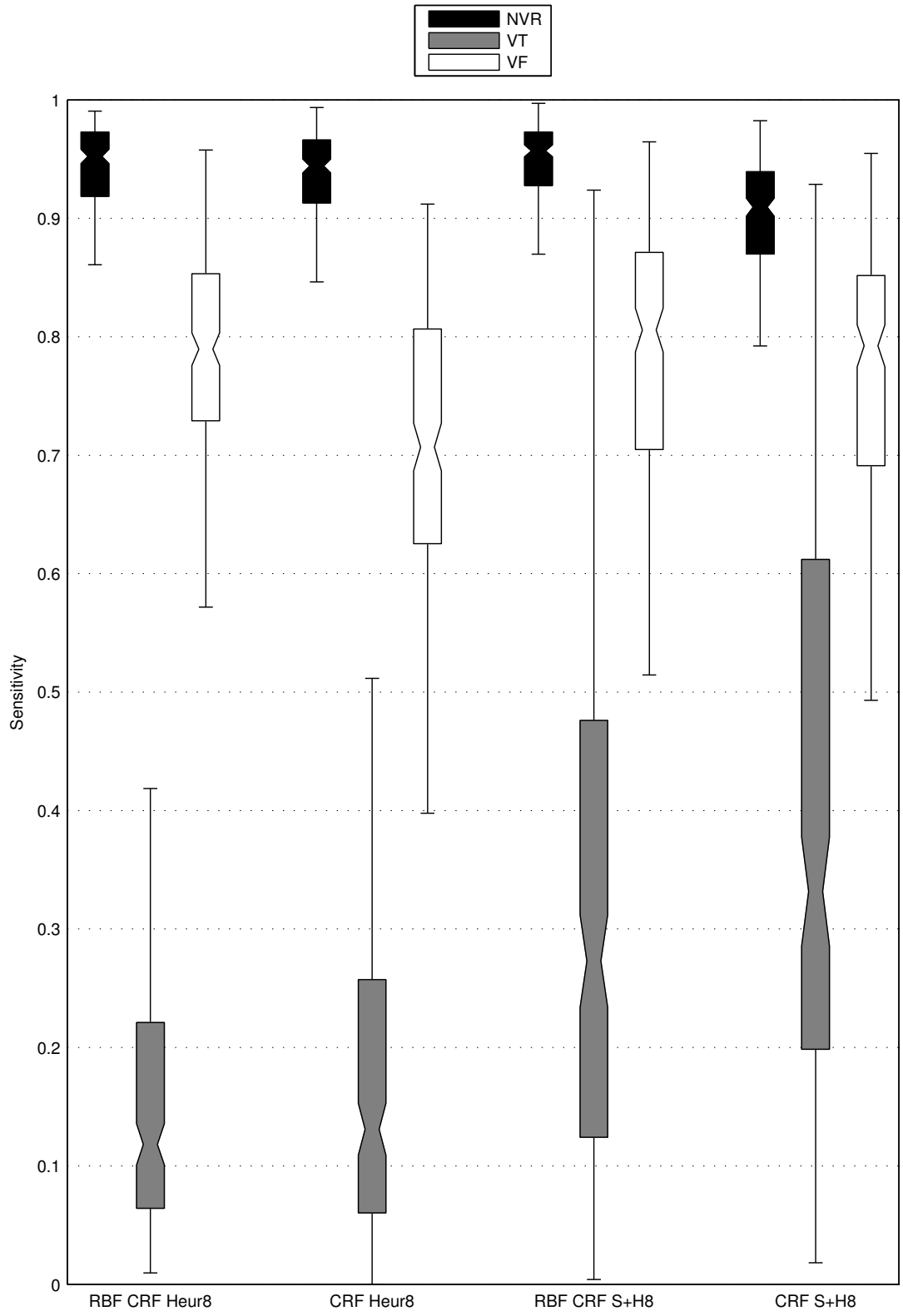


Figure 5.2: Distributions of sensitivities for each of NVR, VT and VF for CRFs trained over RBF SVM outputs,and trained directly over Heur8 and S+H8 representation spaces

that all methods but CRFs trained over S+H8 features obtained median NVR sensitivities competitive with overall best methods from the previous chapter, around 95%. In particular CRFs trained over SVM outputs for the S+H8 representation resulted in competitive NVR and VF sensitivities, but uncompetitive VT sensitivity. The best VT sensitivities were obtained by CRFs trained over S+H8 features directly, with a median sensitivity of 33%.

5.3.3 Discussion

A basic preliminary experiment on the ability of CRFs to perform sequence labelling was performed. CRFs were trained over sequences of SVM hyperplane distance outputs, as well as directly over sequences of considered features. Balanced accuracy results shown from Figure 5.1 were disappointing. An interesting point to note, however, is that the lower whisker of the boxplot showing the distribution of Acc_{bal} scores is substantially lower than that of the Heur8 based CRF classifiers, and even those from previous chapters, whilst the upper whisker is near 90%, indicating that in some cases the classifier can perform well. This may be a consequence of not enough iterations for the CRF solver, which considering that it is a convex problem, in theory should reach the best solution with enough iterations. Interestingly, the median accuracy for a CRF trained directly on the S+H8 features is higher than for the CRF trained over SVM outputs, although the notches overlap so this difference cannot be considered significant. It may however be indicative that the best CRF solution is not being obtained in all cases. The choice of 100 iterations for CRF training was motivated by running time, as CRFs are known to be slow to train due to the fact that the inference step is run repeatedly as part of the training procedure [74].

Other issues that may have hampered the performance of CRFs was the lack of any form of tuning of the regularisation parameter which may have been beneficial

to obtaining better generalising CRFs. An observation context duration of 7 seconds was preselected, due to results from the previous chapter suggesting that this was often a good choice. However this parameter may also benefit from tuning in order to enhance performance. Additionally, only linear chain structured CRFs were considered, due to these being the out of the box defaults for the HCRF software. It is however possible to engineer a different graphical structure which may perform better. Finally, latent variable CRFs were not considered, due to long training times and additional parameters that required tuning. Some unreported preliminary investigations with only a few bootstrap resamples suggested little benefit to consideration of the latent variable variants, but there was simply no way to determine whether the preset parameters were simply inadequate. In particular, for real-time ECG analysis, it is necessary to investigate hidden CRFs [80], as these are the only CRF type capable of realtime discrimination, due to the requirement to perform segmentation.

In general, despite the disappointing results obtained in the very limited study, CRFs and sequential learning in general are techniques that probably warrant a considerably more in depth investigation than presented in this chapter. The results hint at potential for sequential learning to obtain good classification performance, but a substantial number of issues need to be tackled in order to have a more definitive answer.

5.4 Summary and conclusions

Methods for sequential labelling were discussed briefly, based on observations from the previous chapter results that tighter integration of methods can produce an improvement in classification ability. The HMM, MEMM and CRF sequential labelling frameworks were discussed briefly, motivating the use of CRFs from the viewpoint that it dealt with theoretical issues and limitations of the other

two approaches. A brief and by no means complete investigation into the ability of CRFs to perform sequence discrimination in the ECG was conducted using the benchmark representation space, Heur8, and the combination of Spectra and Heur8 features. Despite disappointing results, the upper bound of CRFs trained with the expanded feature space, Spectra and Heur8, suggested that good results may be obtainable, given more computational resources in order to deal with the uncontrolled parameters. Additionally, CRFs provide an unprecedented amount of design flexibility that was simply not exploited in the current experiments.

Chapter 6

Concluding remarks

The thesis is concluded in this chapter. The work conducted in this thesis and the observations resulting from experiments are recalled in Section 6.1, and further considerations are discussed in Section 6.2, including limitations, further experiments to conduct, and recommendations for future research directions.

6.1 Thesis summary

First, in Chapter 1, it was pointed out that cardiovascular diseases are among the leading causes of death. The basics of the electrical conduction system of the heart was described, and how this is expected to map to an observed ECG signal. Examples of three rhythms, SR, VT and VF were shown. The NVR category was clarified as including all rhythms that were not of ventricular origin. Then attention was drawn to the fact that even clinicians may struggle to differentiate between VT and VF, and yet this is an important distinction to make, since the two rhythms respond differently to interventions. Despite recent guidance on how to better discriminate between these rhythms in the ECG, it was not easily transformed into rhythm detection algorithms. This was therefore chosen as the

problem of study for this thesis.

Then, in Chapter 2, specific technical details for understanding the problem and existing approaches in the literature were discussed. First, an overview of representation spaces developed in studies were presented, grouping them by the types of features they derive; temporal, spectral or other types (dynamical, complexity, etc). A summary of the limitations in the way the studies were conducted was then discussed, including experimental flaws, and the low dimension of features derived in all studies. Next, machine learning techniques were discussed, including supervised learning and the most commonly used technique in this thesis, SVMs. Important issues around the use of SVMs for classification were discussed, which included multiclass classification (since the ECG rhythm diagnostics problem as described is a multiclass problem), tunable parameter selection, and dealing with unbalanced data as is the case with the ECG datasets selected. Techniques for assessing models built using classification learners were discussed, along with appropriate metrics for determination of which models perform well. Then, unsupervised learning was touched upon, and PCA for feature dimension reduction was introduced. Finally, details on the ECG databases were provided, including statistics on the rhythms present, selection criteria used for the inclusion of records into the experimental assessment framework, and record preprocessing.

Chapter 3 described two selections of features to be used as benchmarks, from previous studies [3, 15, 19, 25–27, 58], informed by particular studies which performed binary classification between NVR and arrhythmias [32], and between VF and non-VF [31]. Most features were selected from among the highest ranked for the VF and non-VF task, since this has an implicit element of treating VT and VF separately. In order to have higher dimensional features, the Fourier transform of ECG segments was proposed. This was motivated by the fact that taking the absolute value of Fourier spectra also allows to halve the dimension, and enabled

a time-shift insensitive representation. An experimental protocol was designed based on bootstrap resampling, with the goal of properly estimating generalisation capability of each method by ensuring a set of patients was completely unseen by any of the model building procedure. In addition, an important contribution of this work was the use of descriptive statistics, using boxplots to visualise cumulative distribution functions of accuracy scores, which is a departure from typical results presentations in the literature. The goal of the experiments conducted in this chapter was to obtain an idea of what dimension is appropriate, understand the impact of ECG segment length on classification ability, and discover which classification methods and parameters appear to perform the best. Based on the results of this chapter, one of the reference feature spaces was dropped from further consideration, as well as dimension reduction of Fourier spectral features based on PCA, and only the RBF SVM classifier was retained.

In Chapter 4, the goal was to improve upon the best results obtained in Chapter 3. The avenue that was explored for this purpose was through combinations of multiple classifiers in order to improve overall accuracy, as well as building upon the observation that different representation spaces were more capable at discriminating between different groups of ECG rhythm categories. The simplest method, LCEs was effectively performing temporal aggregations of SVM outputs over a given time period in order to exploit expected temporal correlations due to the ECG being a time series. In order to approach the problem in a more principled fashion, the stacked generalisation method was re-purposed for learning a higher level temporal combiner function using machine learning techniques, rather than ad-hoc selection of temporal aggregation functions. Based on the observation that NVR vs arrhythmia, and VT vs VF forms a natural label hierarchy, various approaches to mixing representation spaces were considered, including feature concatenation and hierarchical classifier constructions, in order to try and exploit the apparent strengths of each feature set. Experimental

results showed that combining decisions temporally improved decision accuracy, but combining the representation spaces directly via feature concatenation also produced a substantial boost to diagnostic accuracy. The cumulative improvements when compared with the highest scoring features from [32] was a 12% improvement in overall median accuracy, with the interquartile ranges of the two methods substantially separated.

Finally, supervised sequential labelling was explored inChapter 5. This was motivated by the observation that using previous decision outputs as variables in a higher level predictor function improved the accuracy scores substantially. A brief introduction into the graphical modelling techniques HMMs, MEMMs and CRFs was presented. Due to time constraints, and the fact that the CRF model is the most flexible with the fewest assumptions, only a limited set of CRF models were evaluated, on sequences of features directly, and sequences of SVM outputs. For reasons that were only speculated upon, the CRFs were unable to achieve parity with the temporal ensemble methods constructed in the previous chapter, however the results in this chapter should not be taken as definitive, since the CRFs were not tuned in order to get the best possible results.

6.2 Directions for further research

A new methodology was developed for improving classification between NVR, VT and VF simultaneously in the ECG. Through the process of developing the methodology, some avenues for future work were obvious as a consequence of the developments. On the other hand, observations were also made that are not reported in the thesis. Observations that may directly lead to further research expected to have a positive impact on ECG rhythm classification are discussed in some detail.

6.2.1 Sequential labelling

In Chapter 5, CRFs were briefly explored for developing classifiers utilising information from neighbouring observations. The experiment was not successful, owing to time constraints which influenced a number of factors, including:

1. Insufficient number of iterations used by the CRF solver to reach the global optimum

2. Exploration of only a basic graphical structure underlying the CRF, i.e. interactions between the current label and neighbouring interactions, but no interactions with neighbouring output nodes (labels)

3. No systematic exploration of tunable parameters such as the regularisation value or number of neighbouring interactions, via cross validation

4. More advanced CRF models such as latent variable CRFs [80, 81] were not tested

One aspect to note about CRFs, is that they cannot perform real time diagnostics, due to the way they operate (in fact a direct consequence of solving the label bias problem, see 5.2.2). However, hidden CRFs provide a single output label for the entire sequence. Thus real-time inference can be achieved via segmentation, and it may be possible to replace LDA from the temporal stacking method proposed in Chapter 4 with a hidden CRF. However, an important aspect to explore is the development of the graphical structure of a regular CRF – such a system is still useful for offline diagnostics – since it is noted that engineering the structure of a CRF is to be preferred to using latent variables, as the optimisation problem with latent variables is non-convex [78]. Other possibilities for CRFs include the so-called recurrent CRF [83], and a variation of the hidden CRF based on constrained distributions [82].

Recalling the temporal stacking procedure developed in Chapter 4, it is noted that this technique shares some similarity with, but is not identical to, the recurrent sliding window approach [73], although this was not known at the time of development. Therefore this technique, and others should be explored for sequential labelling to determine how much additional improvement can be obtained by accounting for neighbouring interactions. Possible techniques for this include convolutional neural networks [84], and recurrent neural networks with long short term memory [85]. Particular attention should be paid to convolutional neural networks, since the input and lower hidden layers are convolutional, which makes it seem a good fit for classification of time series data, whilst simultaneously performing feature discovery by the way of learning appropriate filters at the convolutional layers.

6.2.2 Development of features for ECG rhythm classification

An aspect of building classification and learning systems that is often overlooked is the engineering and dimension of the input features. From Chapter 3, it was already clear that higher dimensional features were useful for improving diagnostic accuracy overall. The problem with the derived spectral features was that classification using these features reduced detection sensitivity of NVR, a critical aspect of the problem, since, if NVR are not correctly classified as NVR, they they are being classified as either VT or VF, which would result in an undesirable shock treatment in AED and AICD scenarios. On the other hand, features derived in previous studies obtained good NVR sensitivity (this aspect was often highlighted in previous studies, as defibrillating when no arrhythmia is present can itself induce VF), but at the expense of differentiating between VT and VF.

Recalling Chapter 4 again, the best result presented in this thesis was obtained through the combination of reference features and proposed spectral features, which when combined in certain structures exploiting the label taxonomy allowed for classifiers that improved sensitivities of all categories when compared with previous work. This suggests that further research on feature engineering needs to occur, in order to obtain features good for VT and VF separation, as well as for correctly separating NVR from arrhythmias. Although convolutional neural networks and deep auto encoders have the promise to learn features directly, this should not be relied on and feature engineering should remain an active area of research. In particular, different types of sequential features may be derived, using ECG landmark points, e.g. [86,87]. The curve length transform [86] in particular may be useful, as the curve length becomes large where rapid changes occur, e.g. QRS complex, or erratic ventricular activity. However, these approaches require the use of landmark isolating algorithms, which is not an easy problem in general, and were avoided in this thesis through the use of Fourier transform in order to achieve some form of shift invariance.

6.2.3 Mislabelling in the ECG databases

The problem of differentiating NVR rhythms from VT and from VF was approached in this thesis by the way of supervised learning. This means that rhythms were provided with ground truths, as a gold standard. The rhythm labels were used to estimate a mapping function from a feature space to a decision about which rhythm is currently observed in the ECG. An aspect that was not touched upon was the quality of ECG rhythm labelling. However, given the difficulty in getting experts to agree on some examples of labelled ECGs [2], the existence of the Lambeth Conventions articles [10,11], and a study where human experts had difficulty in classifying some of the evaluation traces [3], it is clear

that one element of the problem which is critical to achieving success in building an automated rhythm detector is to have reliable gold standards for both development and assessment of the algorithms. Of particular note, is the fact that a more recent study [31] relabelled portions of NVRs as VT in the CUDB[1]. However, these new annotations are not, as of the time of writing, available to the public, but acknowledge a critical problem with the gold standard that potentially limits the upper bound on achievable accuracy of automated diagnostics.

Figure 6.1 shows some examples found in the databases of VF that were wrongly labelled as VT, where each episode is 15 seconds long. Figure 6.2 shows 15 second long examples of VT labelled incorrectly as VF. The records are shown from the onset of the rhythms, i.e. the rhythm preceding onset is NVR in all cases. The VT rhythms labelled as VF go on to become VF later in the episode, which would make this labelling consistent with the Lambeth Conventions understanding of rhythm labelling[2]. This reveals a fundamental limitation with the Lambeth Conventions definitions – real time analysis (by either automated diagnostics or a human expert) cannot "foresee" the upcoming VF. This is a technical issue with the definitions – all rhythms are to be labelled as the worst identified in the episode, for the entire episode. This difficulty can be resolved by a small proposed amendment to the definitions, that is, to say that a VT episode may be upgraded to a VF classification *from the point which VF occurs*, and the episode may not subsequently be downgraded to VT.

Therefore, it is clear from both an experimental and observational standpoint that re-annotation of the databases used in this work is essential for further development of automated rhythm classifiers. The challenges for such an effort include developing a collection framework to enable many expert cardiologists (under guidance of Lambeth Conventions to improve consistency) to annotate

[1]Clarified via personal communications with the authors
[2]As clarified via personal communication with Michael Curtis

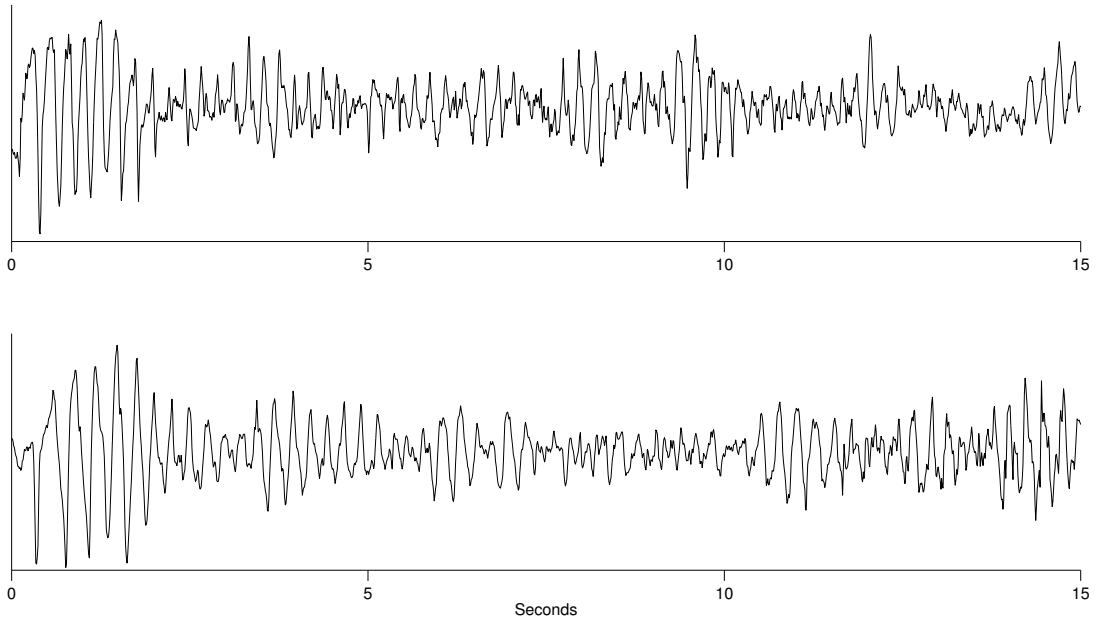


Figure 6.1: Example of VF wrongly labelled as VT

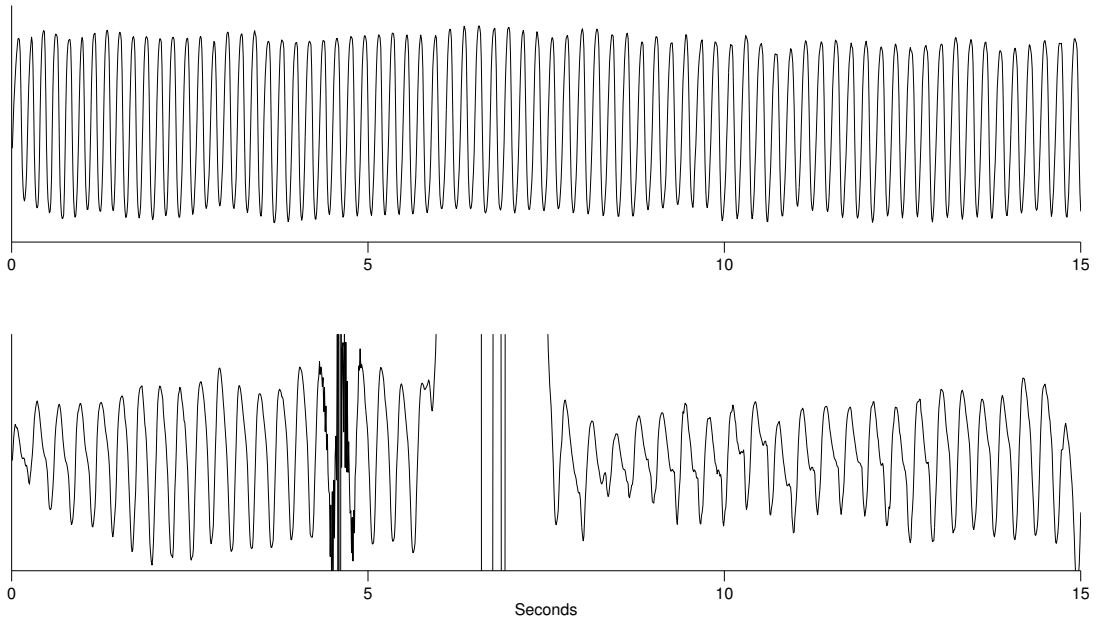


Figure 6.2: Example of VT wrongly labelled as VF

the ECG databases according to NVR, VT and VF labels. A method would be required to obtain final labels from many possibly non-agreeing labels. Alternatively, entirely new classification methods may be developed to learn decision functions in the presence of multiple labels, rather than a single ground truth. This would be an enhancement related to multi-target regression, although the goal would not be to produce multiple labels given an observation. Re-labelling of the databases would enable more consistent learning of decision functions, and additionally, training would benefit from the inclusion of records excluded due to the presence of ventricular flutter labels. Such annotations and the procedure for obtaining them would need to be made public for peer review, and to allow others to reproduce results obtained with the new labelling.

6.3 Final remarks

The problem of differentiating between NVR, VT and VF using only the ECG was studied under a realistic assessment framework. The new experimental framework was essential for understanding the weaknesses of existing methods and ensuring unbiased, realistic assessment of methods for forming diagnostic systems. Methods were developed to improve the diagnostic accuracy simultaneously between all three categories by using higher dimensional and combined feature spaces, and using contextual information from the ECG, rather than considering segments in isolation. The cumulative impact was to reduce overall error rates by 30%, however much more work is required in order to achieve acceptable accuracy in discriminating between VT and VF, particularly given that the co-incidence of VT and VF is high [88].

Future directions for research include discovery of feature spaces that better differentiate between VT and VF, sequential learning methods for exploiting ECG time series interactions, and development of reliable database annotations. It is of

utmost importance that development of methods continues to improve upon NVR sensitivities, whilst reducing incidence of false positives for the AED and AICD scenarios. In the AICD setting this might be achieved through incorporating a patient adaptive beat detection method, e.g. [89], which would be useful for reducing inappropriate shocks [8].

References

[1] W. H. Organization, "WHO — the top 10 causes of death," http://www.who.int/mediacentre/factsheets/fs310/en/, 2014, [Online; accessed 09-December-2014].

[2] R. Clayton, A. Murray, P. Higham, and R. Campbell, "Self-terminating ventricular tachyarrhythmias - a diagnostic dilemma?" *The Lancet*, vol. 341, no. 8837, pp. 93–95, 1993.

[3] I. Jekova and V. Krasteva, "Real time detection of ventricular fibrillation and tachycardia," *Physiological measurement*, vol. 25, no. 5, pp. 1167–1178, Oct 2004.

[4] W. Olson, D. Peterson, L. Ruetz, B. Gunderson, and M. Fang-Yen, "Discrimination of fast ventricular tachycardia from ventricular fibrillation and slow ventricular tachycardia for an implantable pacer-cardioverter-defibrillator," in *Computers in Cardiology*, September 1993, pp. 835–838.

[5] K. Balasundaram, S. Masse, K. Nair, T. Farid, K. Nanthakumar, and K. Umapathy, "Wavelet-based features for characterizing ventricular arrhythmias in optimizing treatment options," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, September 2011, pp. 969–972.

[6] H. Clements-Jewery, D. Hearse, and M. Curtis, "Phase 2 ventricular arrhythmias in acute myocardial infarction: a neglected target for therapeutic antiarrhythmic drug development and for safety pharmacology evaluation," *British Journal of Pharmacology*, vol. 145, no. 5, pp. 551–564, Jul 2005.

[7] M. Chang, E. de Lange, G. Calmettes, A. Garfinkel, Z. Qu, and J. Weiss, "Pro- and antiarrhythmic effects of ATP-sensitive potassium current activation on reentry during early afterdepolarization-mediated arrhythmias," *Heart Rhythm*, vol. 10, no. 4, pp. 575–582, Apr 2013.

[8] J. van Rees, C. Borleffs, M. de Bie, T. Stijnen, L. van Erven, J. Bax, and M. Schalij, "Inappropriate implantable cardioverter-defibrillator shocks: incidence, predictors, and impact on mortality," *Journal of the American College of Cardiology*, vol. 57, no. 5, pp. 556–562, Feb 2011.

[9] C. Stables and M. Curtis, "Development and characterization of a mouse in vitro model of ischaemia-induced ventricular fibrillation," *Cardiovascular Research*, vol. 83, no. 2, pp. 397–404, Jul 2009.

[10] M. J. A. Walker, M. J. Curtis, D. J. Hearse, R. W. F. Campbell, M. J. Janse, D. M. Yellon, S. M. Cobbe, S. J. Coker, J. B. Harness, D. W. G. Harron, A. J. Higgins, D. G. Julian, M. J. Lab, A. S. Manning, B. J. Northover, J. R. Parratt, R. A. Riemersma, E. Riva, D. C. Russell, D. J. Sheridan, E. Winslow, and B. Woodward, "The lambeth conventions: guidelines for the study of arrhythmias in ischaemia, infarction, and reperfusion," *Cardiovascular Research*, vol. 22, no. 7, pp. 447–455, 1988.

[11] M. J. Curtis, J. C. Hancox, A. Farkas, C. L. Wainwright, C. L. Stables, D. A. Saint, H. Clements-Jewery, P. D. Lambiase, G. E. Billman, M. J. Janse, M. K. Pugsley, G. A. Ng, D. M. Roden, A. J. Camm, and M. J. Walker, "The lambeth conventions (ii): Guidelines for the study of animal and human ventricular and supraventricular arrhythmias," *Pharmacology & Therapeutics*, vol. 139, no. 2, pp. 213–248, 2013.

[12] J. Ruiz, E. Aramendi, S. Ruiz de Gauna, A. Lazkano, L. Leturiondo, and J. Gutierrez, "Distinction of ventricular fibrillation and ventricular tachycardia using cross correlation," in *Computers in Cardiology*, September 2003, pp. 729–732.

[13] N. Thakor, Y.-S. Zhu, and K.-Y. Pan, "Ventricular tachycardia and fibrillation detection by a sequential hypothesis testing algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 9, pp. 837–843, September 1990.

[14] C. Zhang, J. Zhao, J. Tian, F. Li, and H. Jia, "Support vector machine for arrhythmia discrimination with TCI feature selection," in *IEEE 3rd International Conference on Communication Software and Networks*, May 2011, pp. 111–115.

[15] M. Arafat, A. Chowdhury, and M. Hasan, "A simple time domain algorithm for the detection of ventricular fibrillation in electrocardiogram," *Signal, Image and Video Processing*, vol. 5, no. 1, pp. 1–10, 2011.

[16] A. Amann, R. Tratnig, and K. Unterkofler, "Reliability of old and new ventricular fibrillation detection algorithms for automated external defibrillators," *BioMedical Engineering OnLine*, vol. 4, no. 1, p. 60, 2005.

[17] L. Khadra, A. Al-Fahoum, and S. Binajjaj, "A quantitative analysis approach for cardiac arrhythmia classification using higher order spectral techniques," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 11, pp. 1840–1845, Nov 2005.

[18] M. A. Othman, N. M. Safri, I. A. Ghani, and F. K. C. Harun, "Characterization of ventricular tachycardia and fibrillation using semantic mining," *Computer and Information Science*, vol. 5, no. 5, pp. 35–44, 2012.

[19] S. Barro, R. Ruiz, D. Cabello, and J. Mira, "Algorithmic sequential decision-making in the frequency domain for life threatening ventricular arrhythmias and imitative artefacts: a diagnostic system," *Journal of Biomedical Engineering*, vol. 11, no. 4, pp. 320–328, 1989.

[20] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[21] B. Bai and Y. Wang, "Ventricular fibrillation detection based on empirical mode decomposition," in *5th International Conference on Bioinformatics and Biomedical Engineering*, May 2011, pp. 1–4.

[22] K. Minami, H. Nakajima, and T. Toyoshima, "Real-time discrimination of ventricular tachyarrhythmia with Fourier-transform neural network," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 2, pp. 179–185, Feb 1999.

[23] A. Lempel and J. Ziv, "On the complexity of finite sequences," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 75–81, September 2006.

[24] X.-S. Zhang, Y.-S. Zhu, N. Thakor, and Z.-Z. Wang, "Detecting ventricular tachycardia and fibrillation by complexity measure," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 5, pp. 548 –555, May 1999.

[25] H. Li, W. Han, C. Hu, and M.-H. Meng, "Detecting ventricular fibrillation by fast algorithm of dynamic sample entropy," in *IEEE International Conference on Robotics and Biomimetics*, Dec 2009, pp. 1105–1110.

[26] A. Amann, R. Tratnig, and K. Unterkofler, "Detecting ventricular fibrillation by time-delay methods," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 1, pp. 174–177, Jan 2007.

[27] ——, "A new ventricular fibrillation detection algorithm for automated external defibrillators," in *Computers in Cardiology*, Sept 2005, pp. 559–562.

[28] Y. Wang, Y.-S. Zhu, N. Thakor, and Y.-H. Xu, "A short-time multifractal approach for arrhythmia detection based on fuzzy neural network," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 9, pp. 989–995, September 2001.

[29] G. Wang, H. Huang, H. Xie, Z. Wang, and X. Hu, "Multifractal analysis of ventricular fibrillation and ventricular tachycardia," *Medical Engineering & Physics*, vol. 29, no. 3, pp. 375–379, 2007.

[30] Y. Li, J. Bisera, M. Weil, and W. Tang, "An algorithm used for ventricular fibrillation detection without interrupting chest compression," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 1, pp. 78–86, Jan 2012.

[31] Q. Li, C. Rajagopalan, and G. Clifford, "Ventricular fibrillation and tachycardia classification using a machine learning approach," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 6, pp. 1607–1613, June 2014.

[32] F. Alonso-Atienza, E. Morgado, L. Fernandez-Martinez, A. Garcia-Alberola, and J. Rojo-Alvarez, "Detection of life-threatening arrhythmias using feature selection and support vector machines," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 3, pp. 832–840, March 2014.

[33] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[34] T. Hastie and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., ser. Springer Series in Statistics. Springer, 2009.

[35] B. D. Ripley, *Pattern recognition and neural networks*. Cambridge university press, 1996.

[36] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 1999.

[37] T. Hastie, A. Buja, and R. Tibshirani, "Penalized discriminant analysis," *The Annals of Statistics*, vol. 23, no. 1, pp. pp. 73–102, 1995.

[38] Y. Lee, Y. Lin, and G. Wahba, "Multicategory support vector machines," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 67–81, 2004.

[39] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 265–292, March 2002.

[40] U. Dogan, T. Glasmachers, and C. Igel, "Fast training of multi-class support vector machines," Faculty of Science, University of Copenhagen, Tech. Rep., 2011.

[41] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, Aug 2011.

[42] J. C. Platt, N. Cristianini, and J. Shawe-taylor, "Large margin DAGs for multiclass classification," in *Advances in Neural Information Processing Systems*. MIT Press, 2000, pp. 547–553.

[43] R. Tibshirani and T. Hastie, "Margin trees for high-dimensional classification," *Journal of Machine Learning Research*, vol. 8, pp. 637–652, May 2007.

[44] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, 1995.

[45] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: a unifying approach for margin classifiers," *Journal of Machine Learning Research*, vol. 1, pp. 113–141, September 2001.

[46] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Machine Learning: ECML 2004*, ser. Lecture Notes in Computer Science, J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, Eds. Springer Berlin Heidelberg, 2004, vol. 3201, pp. 39–50.

[47] A. Cotter, N. Srebro, and J. Keshet, "A GPU-tailored approach for training kernelized SVMs," in *International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 805–813.

[48] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet : Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[49] A. Taddei, G. Distante, M. Emdin, P. Pisani, G. B. Moody, C. Zeelenberg, and C. Marchesi, "The European ST-T database: standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography," *European Heart Journal*, vol. 13, no. 9, pp. 1164–1172, 1992.

[50] F. Nolle, F. Badura, J. Catlett, R. Bowser, and M. Sketch, "CREI-GARD, a new concept in computerized arrhythmia monitoring systems," in *Computers in Cardiology*, vol. 13, 1986, pp. 515–518.

[51] G. Moody and R. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, June 2001.

[52] S. Greenwald, "Development and analysis of a ventricular fibrillation detector," Master's thesis, MIT Dept. of Electrical Engineering and Computer Science, 1986.

[53] ECRI Institute, "American Heart Association ECG database DVD," https://www.ecri.org/Products/Pages/AHA_ECG_DVD.aspx, 2014, [Online; accessed 09-December-2014].

[54] R. Clayton, A. Murray, and R. Campbell, "Changes in the surface ecg frequency spectrum during the onset of ventricular fibrillation," in *Computers in Cardiology*, September 1990, pp. 515–518.

[55] B. Raghavendra, D. Bera, A. Bopardikar, and R. Narayanan, "Cardiac arrhythmia detection using dynamic time warping of ECG beats in e-healthcare systems," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, June 2011, pp. 1–6.

[56] Y. Alwan, Z. Cvetković, and M. J. Curtis, "High-dimensional discriminant analysis of human cardiac arrhythmias," in *European Signal Processing Conference*, Sept 2013, pp. 1–5.

[57] Y. Alwan, Z. Cvetković, and M. Curtis, "Classification of human ventricular arrhythmia in high dimensional representation spaces," *arXiv preprint arXiv:1312.5354*, 2013.

[58] S. Kuo and R. Dillman, "Computer detection of ventricular fibrillation," in *Computers in Cardiology*, 1978, pp. 347–349.

[59] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004.

[60] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[61] T. Shi, M. Belkin, and B. Yu, "Data spectroscopy: Eigenspaces of convolution operators and clustering," *The Annals of Statistics*, vol. 37, no. 6B, pp. 3960–3984, Dec 2009.

[62] W. A. L. Robert McGill, John W. Tukey, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978.

[63] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[64] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, vol. 96, 1996, pp. 148–156.

[65] P. M. Long and R. A. Servedio, "Random classification noise defeats all convex potential boosters," *Machine Learning*, vol. 78, no. 3, pp. 287–304, 2010.

[66] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[67] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, August 1998.

[68] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, vol. 36, no. 6, pp. 1291–1302, 2003.

[69] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[70] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[71] K. M. Ting and I. H. Witten, "Issues in stacked generalization," *Journal of Artificial Intelligence Research*, vol. 10, pp. 271–289, 1999.

[72] T. G. Dietterich, "Machine learning for sequential data: A review," in *International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*. London, UK, UK: Springer-Verlag, 2002, pp. 15–30.

[73] G. Bakiri and T. G. Dietterich, "Achieving high-accuracy text-to-speech with machine learning," in *Data mining in speech synthesis*. Chapman and Hall, 1997.

[74] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[75] C. Sutton and A. McCallum, "An introduction to conditional random fields for relational learning," in *Introduction to statistical relational learning*, L. Geetor and B. Taskar, Eds. MIT press, 2006, pp. 93–128.

[76] A. McCallum, D. Freitag, and F. C. N. Pereira, "Maximum entropy markov models for information extraction and segmentation," in *International Conference on Machine Learning*, 2000, pp. 591–598.

[77] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning*, 2001, pp. 282–289.

[78] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Foundations and Trends in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[79] S. Kumar and M. Hebert, "Discriminative random fields: a discriminative framework for contextual interaction in classification," in *IEEE International Conference on Computer Vision*, October 2003, pp. 1150–1157.

[80] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1848–1852, Oct 2007.

[81] L. Morency, A. Quattoni, and T. Darrell, "Latent-dynamic discriminative models for continuous gesture recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[82] D. Yu, L. Deng, and A. Acero, "Hidden conditional random field with distribution constraints for phone classification," in *Interspeech 2009*. International Speech Communication Association, September 2009.

[83] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, "Recurrent conditional random field for language understanding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014.

[84] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. MIT Press, 1998, pp. 255–258.

[85] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov 1997.

[86] W. Zong, M. Saeed, and T. Heldt, "A QT interval detection algorithm based on ECG curve length transform," in *Computers in Cardiology*, Sept 2006, pp. 377–380.

[87] M. Niknazar, B. Vahdat, and S. Mousavi, "Detection of characteristic points of ECG using quadratic spline wavelet transfrom," in *International Conference on Signals, Circuits and Systems*, November 2009, pp. 1–6.

[88] M. Triventi, S. Valsecchi, M. Landolina, M. Gasparini, M. Lunati, F. Censi, G. Calcagnini, and P. Bartolini, "Analysis of ventricular arrhythmia episodes in patients at risk for ventricular fibrillation," in *Computers in Cardiology*, September 2006, pp. 605–608.

[89] P. De Chazal and R. Reilly, "A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2535–2543, Dec 2006.